Australian Government
Department of Defence
Defence Science and
Technology Organisation

Investigation of Target Motion Analysis in the Presence of Model Uncertainty

T.Q.S. Truong [1] *and D. Koks* [2]

[1] **Maritime Operations Division**

[2] **Electronic Warfare and Radar Division**

Defence Science and Technology Organisation

DSTO–TR–2405

ABSTRACT

This paper presents the results of an investigation of target motion analysis algorithms that are designed to cope with model uncertainty. First, some standard recursive algorithms such as the cartesian extended Kalman filter, modified polar extended Kalman filter, and cartesian unscented Kalman filter are applied to a target motion analysis problem with model uncertainty, in order to analyse the robustness of such algorithms in these conditions. Next, some adaptive algorithms are investigated. They are the static multiple models and the dynamic multiple models estimators, namely: two generalised pseudo Bayes methods and the interacting multiple model method. In this paper, the problem is restricted to a single sensor and a single non-manoeuvring target that travels at constant velocity. Both static and dynamic sensor performances are considered. For simplicity, only Gaussian measurement noise is considered. Adaptive filters are shown to have promise: they can establish a useful bearing standard deviation adaptively and robustly.

Published by
DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh, SA 5111, Australia
Telephone: (08) 8259 5555
Facsimile: (08) 8259 6567

AR No. AR–014–751
May, 2010

APPROVED FOR PUBLIC RELEASE

Investigation of Target Motion Analysis in the Presence of Model Uncertainty

Executive Summary

This work was carried out under the Task 07/093 for the Tracking and Data Fusion subprogram within the Submarine Operations Branch at DSTO Stirling.

In this paper, the effect of model uncertainty in target motion analysis (TMA) performance is investigated. This includes investigating the mismatch of bearing mean and variances between the model and the system. The recursive bearings-only TMA algorithms considered are the cartesian extended Kalman filter (CEKF), modified polar EKF (MPEKF), and cartesian unscented Kalman filter (CUKF). Next, the designs of robust TMA algorithms that are capable of tracking in the presence of model uncertainty are explored. These are adaptive filters, namely the static multiple models (SMM) and the dynamic multiple models estimators. The dynamic multiple models estimators considered in this work are the generalised pseudo Bayes (GPB1 and GPB2) and the interacting multiple model (IMM) algorithms.

The problem considered in this paper is restricted to an ownship with a single sensor observing a single non-manoeuvring constant-velocity target. The ownship must manoeuvre for the algorithms to converge. Three scenarios are described, encompassing low to high rates of bearing change, but only one of these is analysed in detail; the similar results of the others are merely quoted.

The ownship (a submarine) uses passive sonar to detect a target (a ship or a submarine). Two passive sensors onboard submarines that are especially used for tracking are cylindrical and flank arrays. These are located at different places on the submarine, resulting in a lowered performance for each in the relevant zone. The cylindrical array has a *blind arc* behind the submarine from where it receives no data at all. The beamwidth of the flank array is narrow to broadside and broad to endfire (front and behind the submarine), giving poor endfire performance. In this paper, both static and dynamic sensor performances are considered. We use a simple model for the dynamic sensor performance, namely, two fixed bearing standard deviations: one for each half of the scenario. Also, for simplicity, this paper only deals with Gaussian measurement noise.

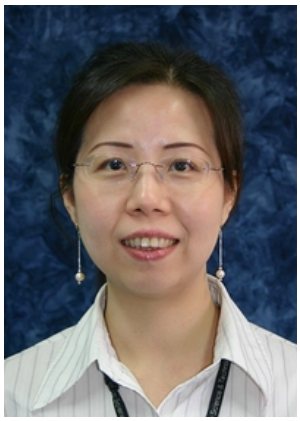
For the problem with a static sensor such as a cylindrical array, results show that the CEKF and the CUKF perform better than the MPEKF. We show that filters incorporating approximately the same bearing mean and error as those of the system outperform filters that are not matched in such a way. This suggests further investigation into *adaptive filters* that estimate these parameters adaptively. In this paper, we focus solely on the design of adaptive filters for eliminating bearing variance mismatch between the model and the system.

The filter used may not "know" the measurement noises (variances and biases). When considering static sensor performance, the adaptive filter (SMM) has been shown to be promising for this kind of problem. The SMM can estimate a bearing standard deviation adaptively, with results comparable to the performance of a filter that uses the same bearing standard deviation as that of the system. For the case of dynamic sensor performance, where the sensor detection ability might be degraded by hardware geometry, there are great benefits to using the GPB1, GPB2, and IMM adaptive filters, because these can estimate a bearing standard deviation adaptively and robustly. We show that they give

comparable performance to that of an ideal filter that uses the dynamic bearing standard deviations of the system. The results obtained from the IMM and GPB2 are closer to those of the ideal filter than the GPB1. The GPB1 gives a marginally smaller position error than the GPB2 and the IMM; but the state covariances calculated using the IMM and GPB2 are more acceptable under the statistical test for consistency. Note that the GPB1 is marginally quicker to process than the IMM, while the GPB2 is easily the slowest. For these reasons, the IMM appears slightly more promising to use.

We conclude that filters using a wrong bearing mean or error can produce a very poor TMA. We show that adaptive filters can give improved performance in the case where the filter has no knowledge of the measurement noise. This is a theoretical investigation and it is proposed to investigate further with real data.

Authors



Tracy Quyen So Truong
Maritime Operations Division

Tracy Truong graduated from Curtin University of Technology with a BSc in Mathematics and Computing in 1994, and a BSc (Hons) in Mathematics (First Class) in 1995. She received her MAppSc in Mathematics at RMIT University in 2005 with a thesis titled "Exploration of a Rendezvous Search Problem using Genetic Algorithms".

She joined DSTO's Air Operations Division in 1996, where she supported the AP-3C, F/A 18, and F-111 operational tactics studies using agent-oriented programming. In 2002 she transferred to Maritime Operations Division, and was involved in studies on effectiveness tactics for the Mk 48 torpedo, assisted in studies on the Mk 46 torpedo defence, and provided exercise analysis tools to the Submarine Operational Analysis Group.

Recently she has been involved in projects for the Submarine Operations Branch, namely target tracking and fusion within the Tracking and Data Fusion subprogram, and assessing the mental workload impact on the submarine Track Manager within the Combat System Design subprogram.

Don Koks
Electronic Warfare and Radar Division

Don Koks completed a doctorate in mathematical physics at Adelaide University in 1996, with a thesis describing the use of quantum statistical methods to analyse decoherence, entropy and thermal radiance in both the early universe and black hole theory. He holds a BSc from the University of Auckland in pure and applied mathematics and physics, and an MSc in physics from the same university with a thesis in applied accelerator physics (proton-induced X ray and γ ray emission for trace element analysis). He has worked on the accelerator mass spectrometry programme at the Australian National University in Canberra, as well as in commercial internet development.

Currently he is a Research Scientist with the Maritime Systems group in the Electronic Warfare and Radar Division at DSTO, specialising in geolocation, geospatial orientation, and weapon–target allocation. He has published a book on mathematical physics called *Explorations in Mathematical Physics: the Concepts Behind an Elegant Language* (Springer, 2006).

Contents

1 Introduction

In this paper, the effect of model uncertainty in bearings-only target motion analysis performance is investigated. Bearings-only target motion analysis (TMA) is vital to a platform's defence. TMA applications can be found in many systems, including sonar systems, airborne warning and control, missile guidance, and anti radiation missiles. Target motion state estimation is used to track a target, avoid collisions, and to evade and attack a target. The use of TMA considered in this report involves a submarine (ownship) tracking a ship or submarine (target) by using passive sonar. The ownship uses a passive system so as not to reveal its presence. Noisy bearings from the sound-radiating target are processed to produce an estimate of the possibly moving target's state (its position and velocity). The use of a passive system means that target range is not explicitly available.

The passive sensors onboard submarines that are particularly used for tracking are the cylindrical array and the flank array [1]. The cylindrical array is located in the submarine's bow, whereas the flank array comprises panels located along the submarine's side. The cylindrical array cannot detect anything behind the submarine. The beamwidth of the flank array is narrow to broadside and broad to endfire (front and behind the submarine), giving poor endfire performance. The flank array is affected by beamsteer direction, while the cylindrical array is not. Thus when the flank array is used for tracking, the beam pattern changes with beamsteer direction; it is narrow when steering to broadside (when the direction of incoming signals is perpendicular to the array, giving the best base line geometry), and broad when steering to endfire (when the incoming direction aligns with the array, giving a poor base line). For this reason, the scenarios investigated in this paper are divided into two classes: static sensor performance (e.g. using a cylindrical array) and dynamic sensor performance (e.g. using a flank array). The dynamic sensor performance considered in this report is a simple one, defined by splitting a scenario into halves. Each half is then assigned a fixed bearing standard deviation.

Since 1960, the Kalman filter (KF) has been one of the most widely used methods for tracking, due to its simplicity, optimality, tractability and robustness. But its application to nonlinear systems can be fraught with difficulty. The most common approach for such systems uses the extended Kalman filter (EKF), which linearises all nonlinear models so that the traditional linear Kalman filter can then be applied. Though widely used, the EKF is difficult to implement and tune; it is only reliable for systems that are almost linear on the update interval [2, 3]. New nonlinear estimators have been developed, such as the unscented Kalman filter (UKF). Unlike the EKF, the UKF does not approximate nonlinear functions; instead, it parametrises means and covariances using a set of sampled points.

Various algorithms have been used for the bearings-only TMA problem [4, 5, 6, 7]. In particular, those used in this paper are the cartesian extended Kalman filter (CEKF) [8, 9], the modified polar EKF (MPEKF) [9, 10], and the cartesian unscented Kalman filter (CUKF) [2, 11, 12]. All of these algorithms are recursive. Unlike batch processing techniques, recursive processing does not require some or all of the previous data to be stored and reprocessed each time a new bearing is measured.

Sonar performance depends on many factors: signal frequency, signal-to-noise ratio, bearing, bearing rate, array type, etc. These factors certainly influence the manufacturer's specification of the expected measurement noise of any particular sensor. Literature studying bearing measurement noise is quite limited. In order to estimate the target state ac-

curately, the estimator needs to know the statistical characteristics of the measurement noise: its bias (mean) and spread (variance). But the above influences mean that this information is not always known, and this leads to the design of *adaptive filters* that can estimate and adjust the measurement noise parameters as new data arrives. In this report we focus on finding the variances of measurement noises adaptively. We are also concerned solely with single sensor/single target tracking.

Next, we explore the designs of adaptive filters that are capable of tracking in the presence of model uncertainty. These are the static multiple models (SMM) and the dynamic multiple models estimators: generalised pseudo Bayes methods (GPB1, GPB2) and the interacting multiple model method (IMM). The algorithms considered are not new but the applications to the problems considered here *are* new. The motivation for using adaptive filters for this problem comes from [13], which shows promising results for using the IMM to track manoeuvring targets whose motion is unknown.

The bearings-only TMA problem considered in this paper assumes a constant-velocity target. To estimate the state of such a target from bearings-only measurements, it is necessary for the ownship to manoeuvre at least once.[1] On the other hand, dealing with motion uncertainty of a target that moves unpredictably in different directions and with variable speeds is much more difficult, and is not considered here.

The class of model uncertainty investigated in this paper is focused on the mismatch of measurement noise between the model and the system, namely:

- Mismatch of the Gaussian means; i.e. the model assumes a zero-mean measurement noise while the system measurement noise is non-zero mean, and
- Mismatch of the Gaussian variances.

In short, the primary investigations considered in this paper centre on these two questions:

- Benchmarking of the algorithms (CEKF, MPEKF, CUKF) to cope with different degrees of mismatch as stated earlier. In other words, is the algorithm robust in this circumstance?
- Can we design adaptive filters to recover from the model mismatch?

This paper is organised as follows.

Section 2 describes the bearings-only TMA problem for a non-manoeuvring target;

Section 3 describes the performance metrics used in our analysis;

Section 4 describes the simulations and parameter values;

Section 5 presents experimental results and discussions for the standard filters (CEKF, MPEKF, CUKF);

Section 6 briefly describes the adaptive filters used in the simulations (SMM, GPB1, GPB2, IMM);

[1] Note that it is not actually *sufficient* for the ownship to manoeuvre in order to estimate the target state accurately. A complicated manoeuvre or set of manoeuvres might be needed, and the accuracy of the state estimate can begin to degrade some time after the last manoeuvre finishes.

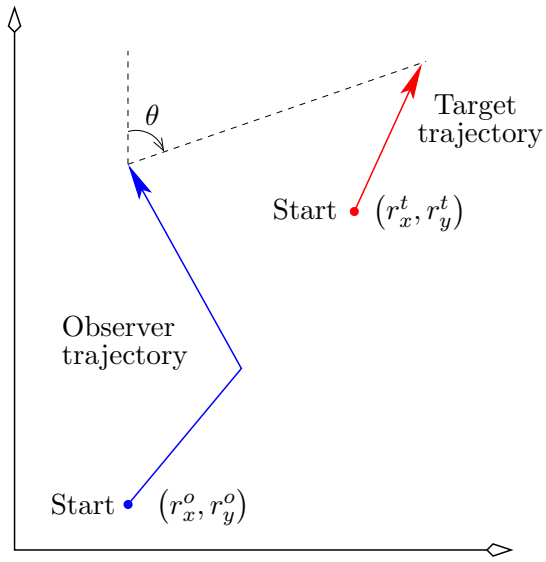


Figure 1: *A typical two-dimensional target–observer geometry*

Section 7 presents experimental results and discussions for the adaptive filters (which only use multiple CEKFs and CUKFs); and finally

Section 8 summarises our key results and provides some recommendations and future research directions.

2 Bearings-Only TMA Problem for a Non-Manoeuvring Target

The basic scenario considered in this paper is shown in Figure 1. This is a classical two-dimensional bearings-only TMA problem comprised of a single non-manoeuvring target and a single manoeuvring observer or ownship whose trajectory consists of two legs, each with a different heading. The target has cartesian position $\left(r_x^t, r_y^t\right)$ and moves with a nearly constant velocity vector $\left(v_x^t, v_y^t\right)$. We write its state vector as

$$\boldsymbol{x}^t = \left[r_x^t, r_y^t, v_x^t, v_y^t\right]^T , \tag{2.1}$$

with superscript T denoting the transpose (i.e. all vectors are columns). Similarly, the ownship state is written

$$\boldsymbol{x}^o = \left[r_x^o, r_y^o, v_x^o, v_y^o\right]^T , \tag{2.2}$$

where the velocity vector is typically constant for each leg. The relative state vector is

$$\boldsymbol{x} \equiv \boldsymbol{x}^t - \boldsymbol{x}^o \equiv \left[r_x, r_y, v_x, v_y\right]^T . \tag{2.3}$$

Bearings-only tracking estimates the target state of a discrete time dynamic process using the least mean squared error criterion. The target's relative state at time index $k = 1, 2, \ldots$ is modelled by

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k - \boldsymbol{U}_{k,k+1} + \boldsymbol{v}_k , \tag{2.4}$$

where the *state transition matrix* for a sampling interval T is

$$\boldsymbol{F}_k = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

Also

$$\boldsymbol{U}_{k,k+1} = \begin{bmatrix} r_x^o(k+1) - r_x^o(k) - Tv_x^o(k) \\ r_y^o(k+1) - r_y^o(k) - Tv_y^o(k) \\ v_x^o(k+1) - v_x^o(k) \\ v_y^o(k+1) - v_y^o(k) \end{bmatrix}, \tag{2.6}$$

and the random variable $\boldsymbol{v}_k$ models the process noise, assumed Gaussian:

$$\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q_k}), \tag{2.7}$$

where

$$\boldsymbol{Q}_k = q \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix}, \qquad q \equiv \text{process noise intensity parameter.} \tag{2.8}$$

The bearing measurements are z_k:

$$z_k = h_k(\boldsymbol{x}_k) + w_k, \tag{2.9}$$

where

$$h_k(\boldsymbol{x}_k) = \text{true bearing } \theta = \tan^{-1}\frac{r_x(k)}{r_y(k)} + \text{quadrant-dependent constant}, \tag{2.10}$$

and the random variable w_k models the measurement noise, again assumed Gaussian:

$$w_k \sim \mathcal{N}\Big(0, \big(\sigma_\theta^2\big)_k\Big). \tag{2.11}$$

3 Performance Metrics

In order to compare the performance of various tracking algorithms, 100 Monte Carlo simulations were run for each algorithm, implemented using Matlab. Each set of runs used the same seed for the random number generator; this ensured that the same set of bearings was used for each set of runs. The use of averaging in the following performance metrics significantly reduced the possibility of a single "lucky" run—one that is not representative of a typical algorithm performance.

The performance metrics that were used in the analysis are as follows. Suppose there are M independent Monte Carlo runs. First, the **root mean square position error** (RMS) at time (index) k is defined as

$$\text{RMS}_k = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\big[r_x(k)^i - \hat{r}_x(k)^i\big]^2 + \big[r_y(k)^i - \hat{r}_y(k)^i\big]^2}, \tag{3.1}$$

where $\big(r_x(k)^i, r_y(k)^i\big)$ and $\big(\hat{r}_x(k)^i, \hat{r}_y(k)^i\big)$ are the true and estimated target positions at time k in the i^{th} Monte Carlo run.

Next, for a particular scenario and parameter choice, the **root time-averaged mean square position error** (RTAMS) is calculated. Unlike the RMS error at time k, the RTAMS takes multiple times into account and so is a more complete indicator of the algorithm's performance. It is defined as

$$\text{RTAMS} = \sqrt{\frac{1}{(t_{\max}-l)M}\sum_{k=l+1}^{t_{\max}}\sum_{i=1}^{M}\big[r_x(k)^i - \hat{r}_x(k)^i\big]^2 + \big[r_y(k)^i - \hat{r}_y(k)^i\big]^2}, \tag{3.2}$$

with $t_{\max}$ the final time of the scenario, and l a time after which the averaging is carried out; hence $t_{\max} - l$ is the interval over which the averaging is carried out. Typically, l is chosen to coincide with the end of the first ownship manoeuvre, or a time slightly later. For example, l was set to the index appropriate for 18 minutes in the scenario of Figure 2 (refer to Table 1 and the discussion at the start of Section 5).

The **norm of the bias position error at the final time** is

$$\frac{1}{M}\sqrt{\left\{\sum_{i=1}^{M} r_x^i - \hat{r}_x^i\right\}^2 + \left\{\sum_{i=1}^{M} r_y^i - \hat{r}_y^i\right\}^2}, \tag{3.3}$$

where $\big(r_x^i, r_y^i\big)$ and $\big(\hat{r}_x^i, \hat{r}_y^i\big)$ are the true and estimated target positions at the final time $t_{\max}$ in the i^{th} Monte Carlo run.

The **average normalised (state) estimation error squared** (NEES) at time k is

$$\overline{\text{NEES}}_k = \frac{1}{M}\sum_{i=1}^{M}\left[\left(\boldsymbol{x}_k^i - \hat{\boldsymbol{x}}_k^i\right)^T \left(\boldsymbol{P}_{k|k}^i\right)^{-1}\left(\boldsymbol{x}_k^i - \hat{\boldsymbol{x}}_k^i\right)\right], \tag{3.4}$$

where $\boldsymbol{x}_k^i$ and $\hat{\boldsymbol{x}}_k^i$ are the true and estimated state vectors $[r_x, r_y, v_x, v_y]^T$ and $[\hat{r}_x, \hat{r}_y, \hat{v}_x, \hat{v}_y]^T$ respectively, and $\boldsymbol{P}_{k|k}^i \equiv \mathbb{E}\left\{(\boldsymbol{x}_k^i - \hat{\boldsymbol{x}}_k^i)(\boldsymbol{x}_k^i - \hat{\boldsymbol{x}}_k^i)^T\right\}$ is the covariance matrix, at time k and i^{th} Monte Carlo run.

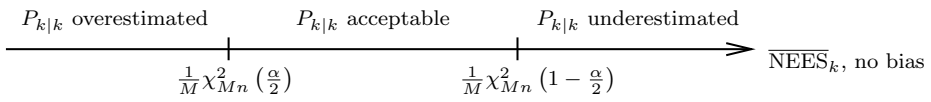
To satisfy the consistency criteria for a filter, the state errors should be zero mean with magnitude commensurate with the state covariance as yielded by the filter. That is, the state estimation errors are consistent with the filter-calculated covariances if $\overline{\text{NEES}}_k$ is within the following χ^2 interval:

$$\frac{1}{M}\left[\chi^2_{Mn}\left(\frac{\alpha}{2}\right),\ \chi^2_{Mn}\left(1-\frac{\alpha}{2}\right)\right], \tag{3.5}$$

where $Mn = 400$ is the number of degrees of freedom for the χ^2 test ($M = 100$ Monte Carlo runs and $n = 4$ is the dimension of the $\boldsymbol{x}$ state vector), and $\alpha/2$ is the "tails part" of the two-sided $(1-\alpha)$-probability concentrate region. (Further details can be found in [14].)

In other words, assuming the state estimate to be unbiased, if $\overline{\text{NEES}}_k$ lies within the interval of (3.5), we deem the state covariance calculated by the filter to be acceptable. Still assuming no bias, if $\overline{\text{NEES}}_k$ is below the lower bound, we infer that the filter has

overestimated the state covariance. Similarly, if $\overline{\text{NEES}}_k$ is above the upper bound, we infer that the filter has underestimated the state covariance:



4 Simulations and Parameter Settings

Figure 2 shows the target–ownship geometry that we call "Scenario 1" and use throughout this report. Ownship and target parameters are listed in Tables 1 and 2. Parameter choices are straightforward, with the possible exception being those describing the ownship manoeuvre. The ownship maintains a constant velocity until a time of 13 minutes, when it changes to a heading of 20° with a turning rate of 0.5°/s. It maintains this new heading until the end of the observation period is reached after 30 minutes. Bearing measurements are received every 20 seconds, with a nominal bearing accuracy of 1.5°. The results presented in the sections to follow are based on this scenario, in which the rate of increase of target bearing as seen by the ownship peaks at 12°/min. Two other scenarios with lower and higher bearing rates (peaking at 2°/min and 40°/min respectively) are shown in Figure 3. Results of these scenarios show trends similar to those of Scenario 1, and so are not analysed further here.

Figure 4 shows typical bearings received versus time for each of the three scenarios. The filters were initialised by the method proposed in Chapter 6 of [11], with settings listed in Table 3. These settings were extracted from [11, 15, 16, 17] for range, speed, course and process noise intensity respectively. For each Monte Carlo run, parameters were chosen according to

$$\begin{aligned}
&\text{initial target detection range } r \sim \mathcal{N}(\bar{r}, \sigma_r^2)\,,\\
&\text{initial target detection speed } s \sim \mathcal{N}(\bar{s}, \sigma_s^2)\,,\\
&\text{initial target detection course } c \sim \mathcal{N}(\bar{c}, \sigma_c^2)\,.
\end{aligned}$$

Other filter initialisations (such as initial state vector and its covariance) were obtained from [11].

Table 1: *Ownship parameters*

Initial (x, y) position	(0,0) km	
Speed	7 kts constant	
Heading	140°	(0 → 13 minutes)
	140° → 20°	(13 → 17 minutes)
	20°	(17 → 30 minutes)
Rate of turn	0.5°/s	
Bearing measurement interval	20 s	
Bearing measurement error σ_θ	1.5°	

Table 2: *Target parameters*

Initial range	10 km
Speed	15 kts constant
Initial true bearing	60°
Heading	195° constant

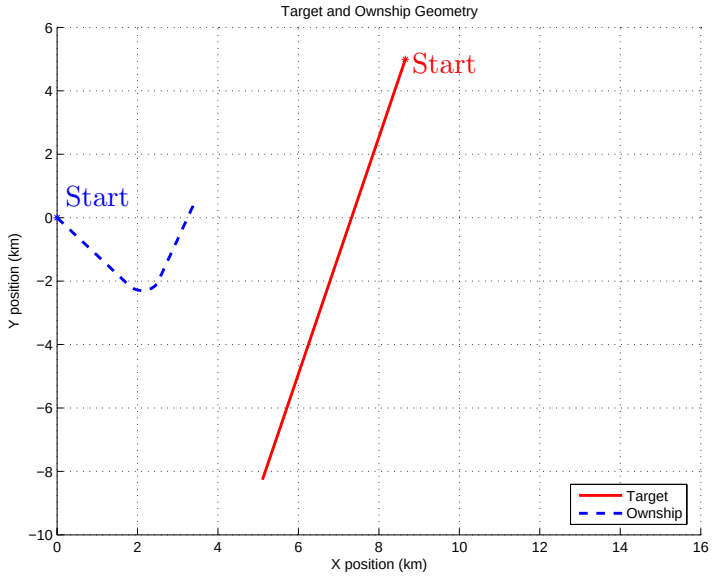


Figure 2: *Scenario 1: target–ownship geometry*

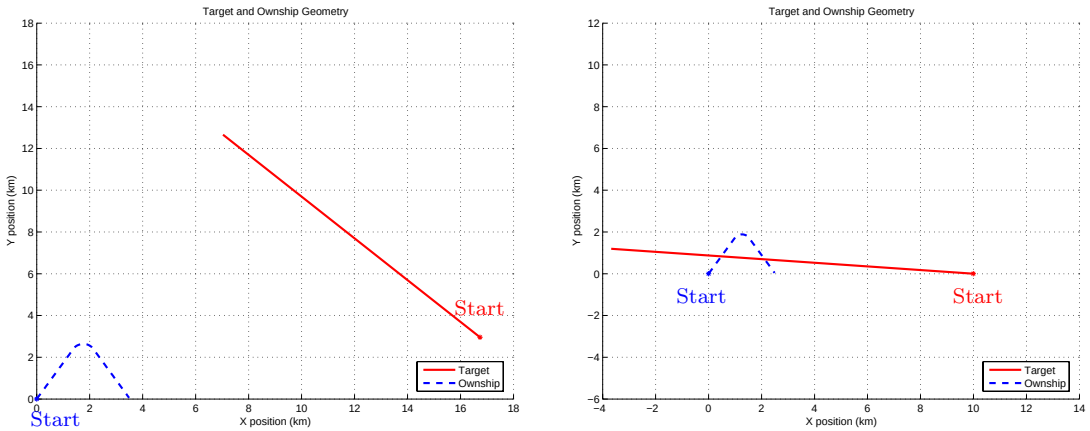


Figure 3: *Scenarios 2 (left) and 3 (right). Their analyses are not presented in this report, but gave results similar to those of Scenario 1*

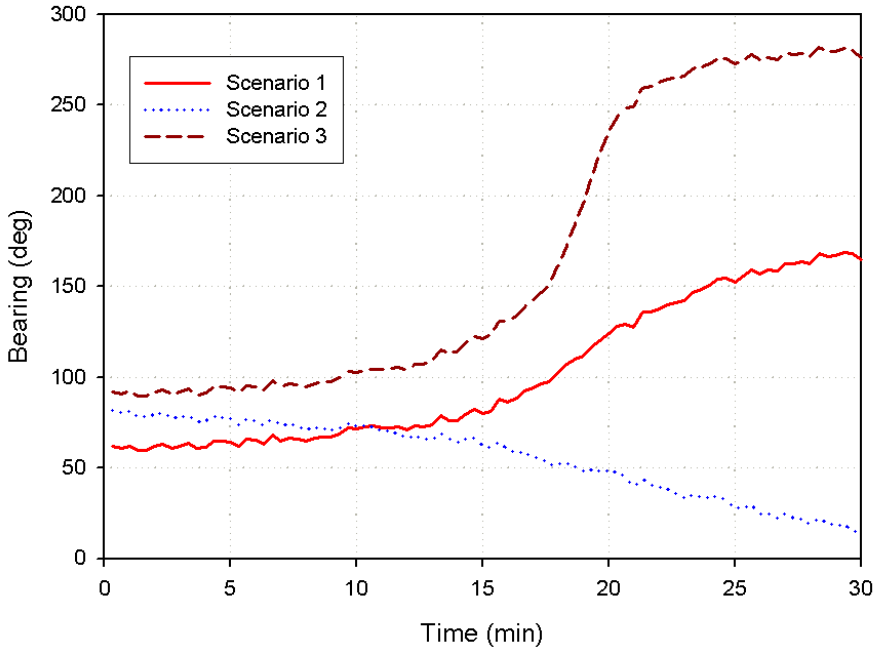


Figure 4: *Target bearings seen by ownship as a function of time for Scenarios 1, 2 and 3*

Table 3: *Nominal filter parameters for a target*

Range mean $\bar{r}$	15 km
Range standard deviation σ_r	6 km
Speed mean $\bar{s}$	16 kts
Speed standard deviation σ_s	7 kts
Course mean $\bar{c}$	$\theta_1 + 180°$, where θ_1 = initial bearing measurement
Course standard deviation σ_c	$180°/\sqrt{12}$
Process noise intensity q	$0.001 \times 10^{-6}\,\mathrm{km}^2/\mathrm{s}^3$

5 Results for the Standard Filter Benchmarks

Presented in this section are the results of a number of Monte Carlo runs aimed at benchmarking the standard algorithms (CEKF, MPEKF, CUKF) in the presence of model mismatch. Parameter settings for the non-manoeuvring target bearings-only TMA problem are used, as described in the previous section.

For a representative run, the target position estimated using CEKF, MPEKF, and CUKF algorithms together with the actual target position are plotted in Figure 5. Here, the system and filters are assumed to have the same bearing measurement noise statistics: zero-mean Gaussian with standard deviation 1.5°. The performance of the filters—RMS position error versus time—is presented in Figure 6. This figure shows that the ownship completes its first manoeuvre after about 17 minutes, and slightly after this is when good convergence to the ground truth begins. (Thus, referring to the discussion just after (3.2), we set $l = 18$ minutes in that equation.) We are interested in the performance results from this time onward, so focus on the results of the parameters described in (3.1)–(3.4). In order to simulate a model mismatch, the following experiments are considered, with results presented in Sections 5.1 and 5.2:

- **Mismatch of the Gaussian variances used in the model and the system (Section 5.1):** Here the system bearing measurement noise is assumed Gaussian with zero mean and varying standard deviation $\sigma_\theta = 1.5°, 2.5°, 3.5°, 4.5°$. These values were chosen partly with reference to [1] and partly from past data. For each σ_θ, the filters (CEKF, MPEKF, CUKF) simulate a bearing noise as Gaussian, each with zero mean and the following standard deviations: $\sigma_\theta/3, \sigma_\theta/2, \sigma_\theta,\ 2\sigma_\theta, 3\sigma_\theta$ and, finally, $10\sigma_\theta$ (to investigate filter performance for a very large bearing standard deviation).
- **Mismatch of the Gaussian means used in the model and the system (Section 5.2):** In this case the system bearing measurement noise is assumed Gaussian with a fixed standard deviation $\sigma_\theta = 1.5°$, but now with a mean varying over the values 0°, 1°, 2°. The filters simulate a bearing noise as Gaussian with zero mean and the same standard deviation as the system ($\sigma_\theta = 1.5°$).

5.1 Mismatch of the Gaussian variances used in the model and the system

This section presents comparison results of the overall performance of each filter (CEKF, MPEKF, CUKF) in the presence of bearing measurement variance mismatch between the model and the system as stated in the immediately foregoing paragraphs.

In order to compare the overall performance of the filters, the following plots present their data as a function of the standard deviation of the filters. Note that the "tick-mark spacing" on the x axis of these plots has been set constant for convenience only, to facilitate presenting the data more compactly.

Figure 7: RMS position error at the final time of $t = 30$ minutes,

Figure 8: RTAMS position error (an average over the $18 \rightarrow 30$ minutes interval),

Figure 9: norm of bias position error at $t = 30$ minutes, and

Figure 10: $\log_{10} \overline{\text{NEES}}$ at $t = 30$ minutes.

Figures 7–9 show that filters using approximately the same bearing standard deviation as that of the system perform better than those using a smaller estimated error. These figures also show that filters using a larger bearing standard deviation than that of the system can perform better than those using the same bearing standard deviation, except in regards to the NEES of Figure 10. We see on a further examination of this figure that the average NEES for the CEKF and CUKF are also within the *acceptable average bounds*.

> The *acceptable average bounds* are defined to be the average NEES bounds obtained from [14], and are used throughout this paper. They specify the interval $\frac{1}{M}\left[\chi^2_{Mn}(\alpha/2),\ \chi^2_{Mn}(1-\alpha/2)\right]$. In our case, $M = 100$ Monte Carlo runs, $n = 4$ counts the dimensions of the $\boldsymbol{x}$ state vector, and $\alpha/2 = 2.5\%$ for the two-sided 95% probability region. The interval of the acceptable average bounds is thus $\frac{1}{100}\left[\chi^2_{400}(0.025),\ \chi^2_{400}(0.975)\right] = [3.46, 4.57]$.

Overall, taking into consideration error performance and NEES, we conclude that a filter is superior if it uses the same bearing standard deviation as that of the system. We also observe that in this case, the state covariances (components of $\overline{\text{NEES}}$) that are calculated using the CEKF and CUKF lie within the appropriate χ^2 bounds, and so are more acceptable than those of the MPEKF, which do not. Reference [18] shows that a filter—such as our MPEKF—that estimates its covariance optimistically is likely to diverge. This suggests that the CEKF and CUKF are more acceptable for our work than the MPEKF.

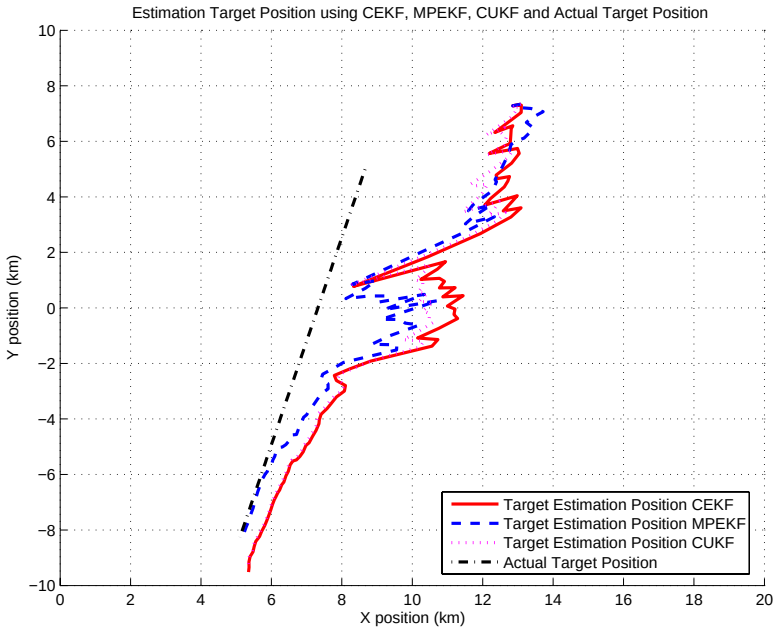


Figure 5: *Estimated target position using CEKF, MPEKF, CUKF, together with actual target position*

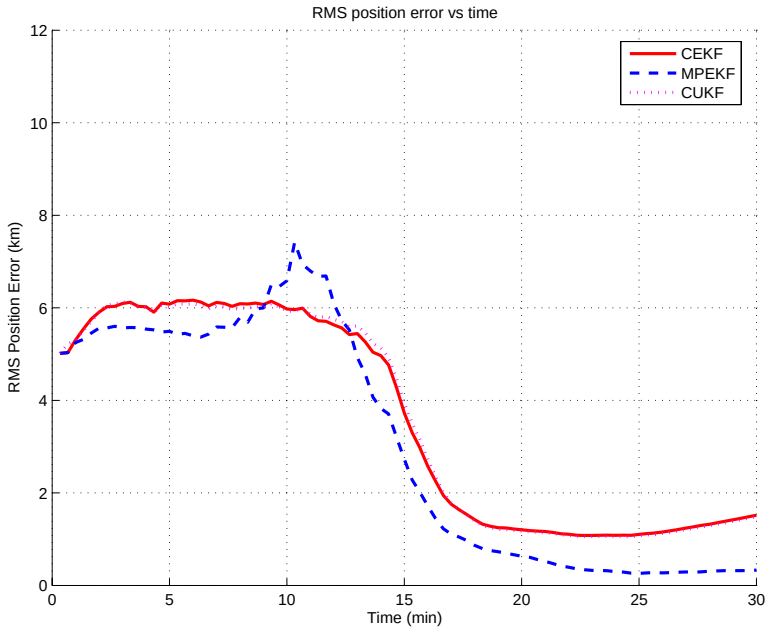
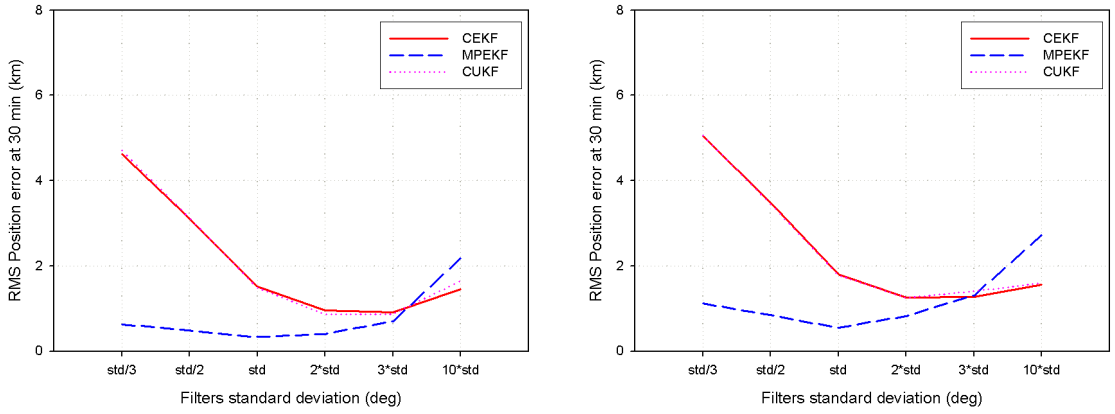
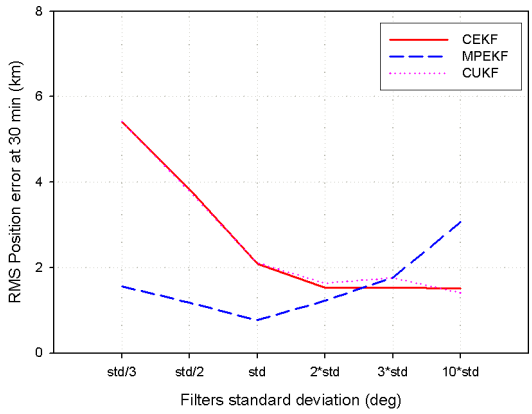


Figure 6: *RMS position errors for CEKF, MPEKF, CUKF*

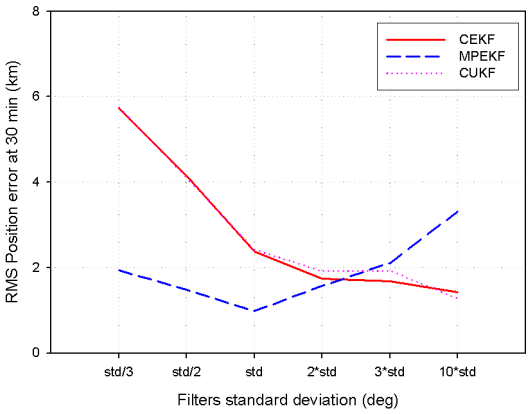


(a) $\sigma_\theta = 1.5°$

(b) $\sigma_\theta = 2.5°$

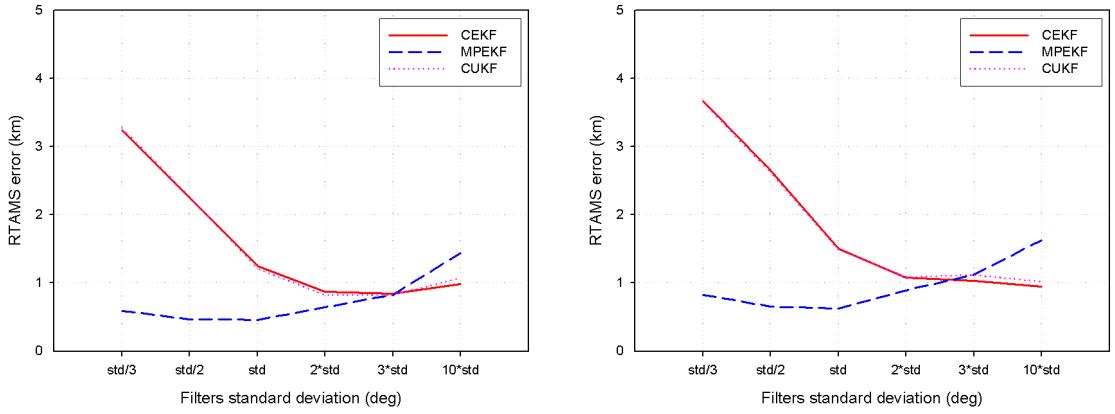


(c) $\sigma_\theta = 3.5°$



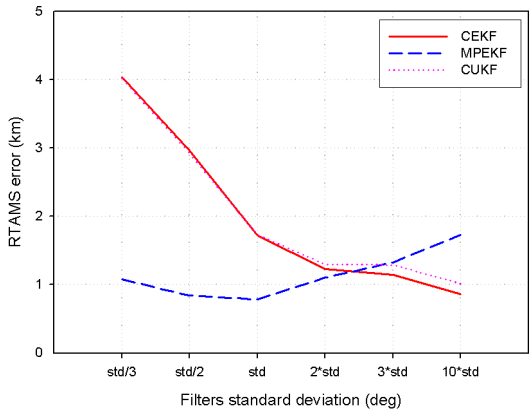
(d) $\sigma_\theta = 4.5°$

Figure 7: *RMS position error at 30 minutes versus filter bearing standard deviation for several system bearing standard deviations σ_θ. Note that "std" in each plot means σ_θ*

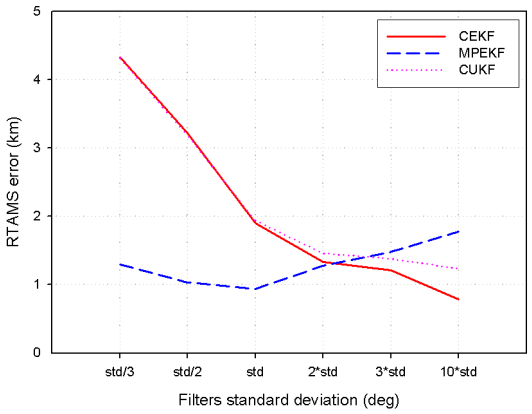


(a) $\sigma_\theta = 1.5°$

(b) $\sigma_\theta = 2.5°$

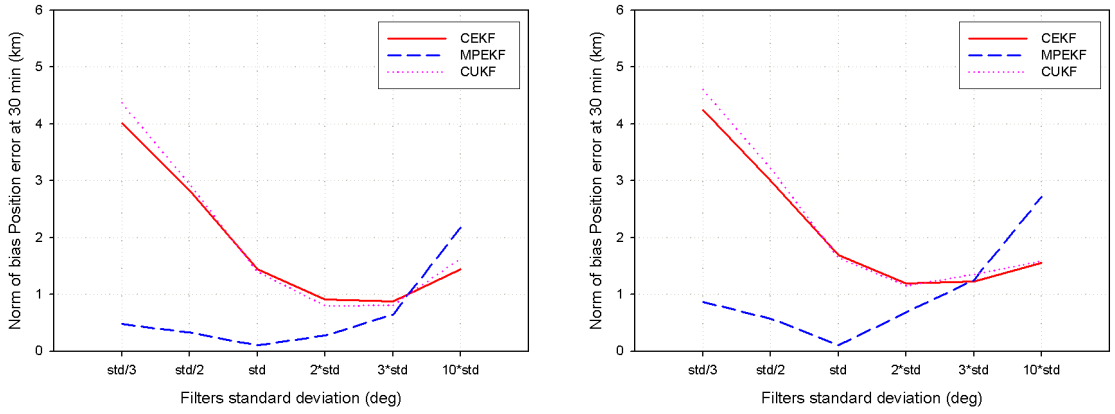


(c) $\sigma_\theta = 3.5°$



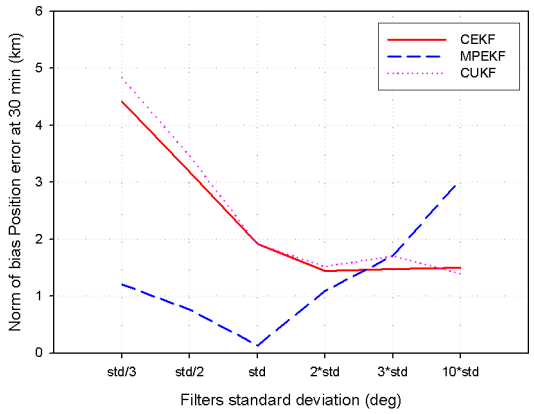
(d) $\sigma_\theta = 4.5°$

Figure 8: *RTAMS position error versus filter bearing standard deviation for several system bearing standard deviations σ_θ. Note that "std" in each plot means σ_θ*

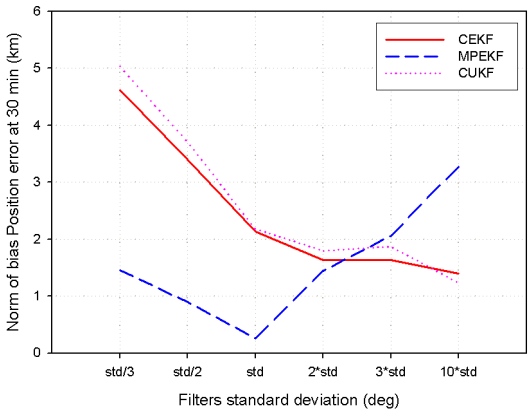


(a) $\sigma_\theta = 1.5°$

(b) $\sigma_\theta = 2.5°$

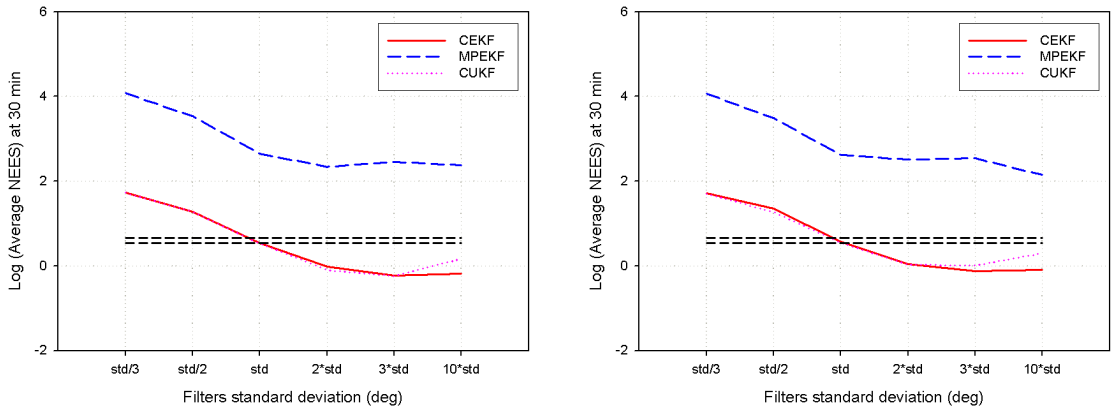


(c) $\sigma_\theta = 3.5°$

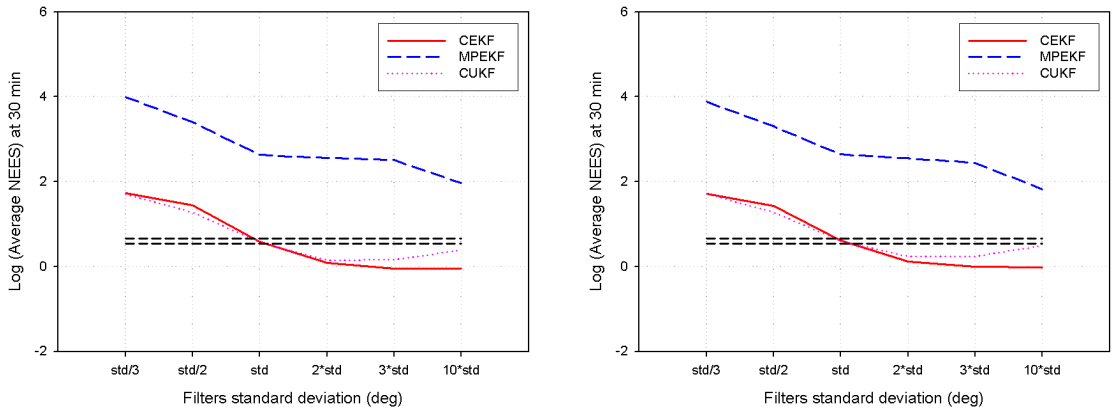


(d) $\sigma_\theta = 4.5°$

Figure 9: *Norm of bias position error at 30 minutes versus filter bearing standard deviation for several system bearing standard deviations σ_θ. Note that "std" in each plot means σ_θ*



(a) $\sigma_\theta = 1.5°$ (b) $\sigma_\theta = 2.5°$



(c) $\sigma_\theta = 3.5°$ (d) $\sigma_\theta = 4.5°$

Figure 10: $\log_{10} \overline{NEES}$ *at 30 minutes versus filter bearing standard deviation for several system bearing standard deviations* σ_θ*. The horizontal black dashed lines across are the logs of the acceptable average bounds. Note that "std" in each plot means* σ_θ

5.2 Mismatch of the Gaussian means used in the model and the system

This section presents comparison results of the overall performance of each filter (CEKF, MPEKF, CUKF) in the presence of bearing measurement mean mismatch between the model and the system as described on page 9. That is, we consider Gaussian system bearing measurement noises with means $\mu = 0°, 1°, 2°$, along with a fixed standard deviation $\sigma_\theta = 1.5°$. The filters simulate a bearing noise as Gaussian with zero mean and the same standard deviation (1.5°) as the system.

In order to compare the overall performance of each filter having a bearing mean mismatch between the model and the system, the following plots are presented:

Figure 11: RMS position error at the final time of $t = 30$ minutes,

Figure 12: RTAMS position error (an average over $18 \rightarrow 30$ minutes interval),

Figure 13: norm of bias position error at $t = 30$ minutes, and

Figure 14: $\log_{10} \overline{\text{NEES}}$ at $t = 30$ minutes.

Overall, the plots show that if the mean of bearing measurement noise of the model is not approximately equal to that of the system, a very poor TMA is obtained. So it is crucial to have a filter that matches the bearing means of the model and system. This suggests further research into filters that can estimate the bearing mean adaptively.

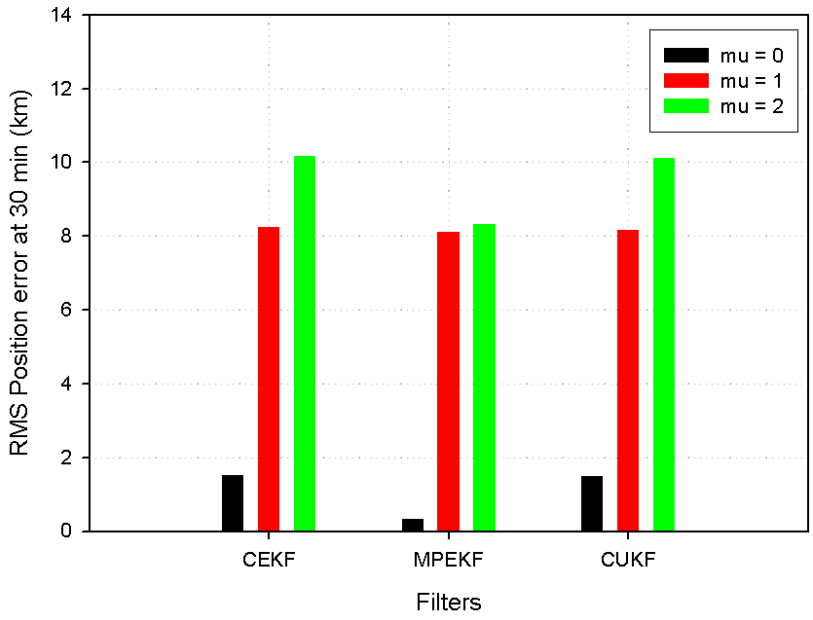


Figure 11: *RMS position error at 30 minutes for filters with a bearing mean of 0°, and system bearing means of 0°, 1°, 2°. Both system and filters have* $\sigma_\theta = 1.5°$

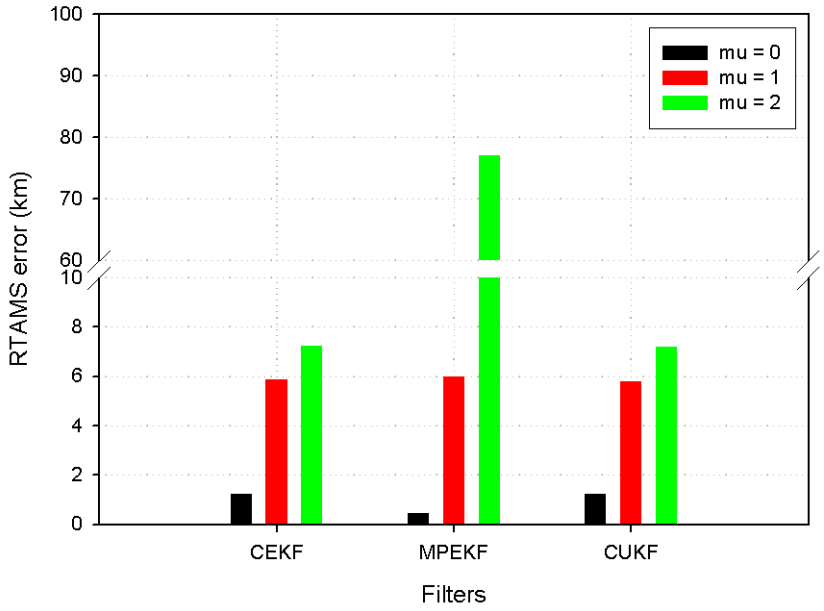


Figure 12: *RTAMS position error for filters with a bearing mean of 0°, and system bearing means of 0°, 1°, 2°. Both system and filters have* $\sigma_\theta = 1.5°$

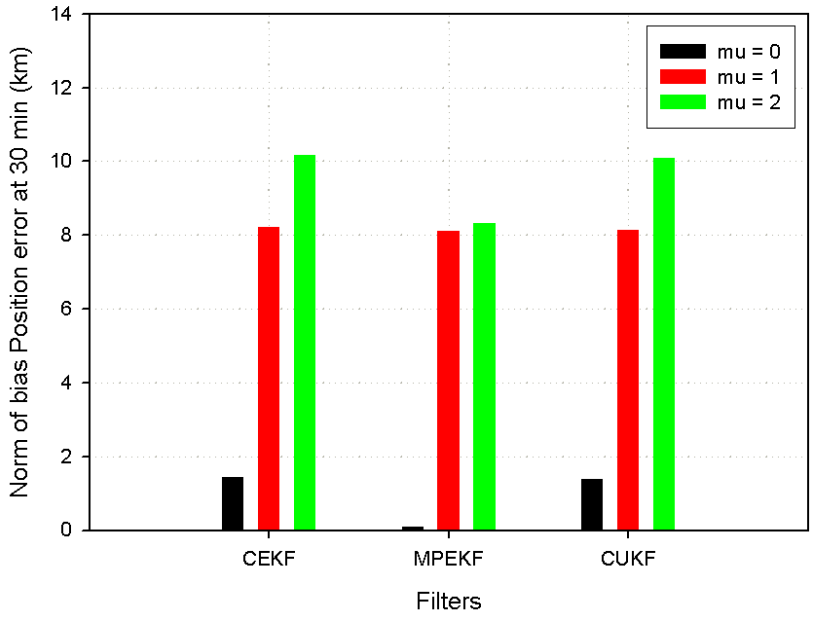


Figure 13: *Norm of bias position error at 30 minutes for filters with a bearing mean of 0°, and system bearing means of 0°, 1°, 2°. Both system and filters have* $\sigma_\theta = 1.5°$

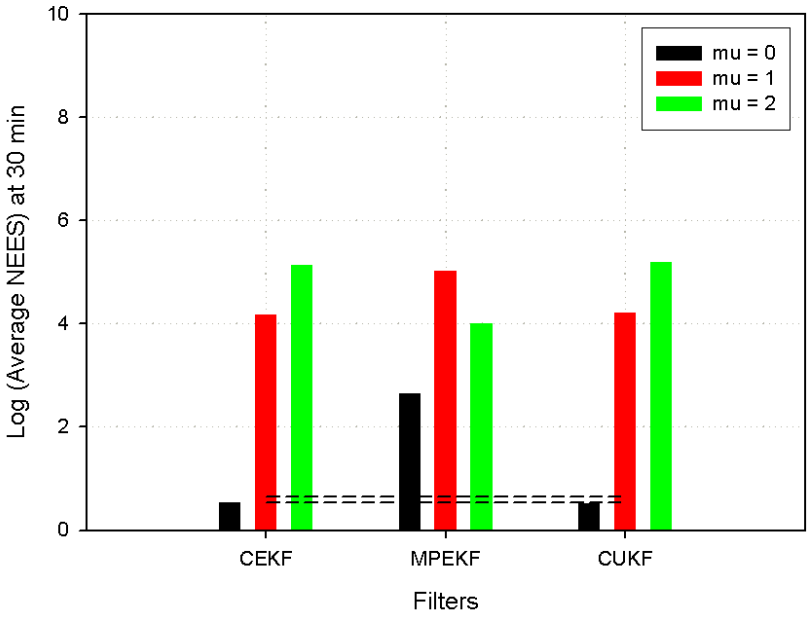


Figure 14: $\log_{10} \overline{NEES}$ *at 30 minutes for filters with a bearing mean of 0°, and system bearing means of 0°, 1°, 2°. Both system and filters have* $\sigma_\theta = 1.5°$*. The horizontal black dashed lines across are the logs of the acceptable average bounds*

6 Adaptive Filters

We first briefly describe the adaptive filters (i.e. multiple models techniques) used in this paper. Further details can be found in [14, 19]. These filters are:

Static multiple models estimator (SMM), and

Dynamic multiple models estimators, namely: Generalised Pseudo Bayes methods (GPB1 and GPB2) and the interacting multiple models method (IMM).

6.1 Static Multiple Models Estimator (SMM)

The system is assumed to obey a fixed one of r possible models, $M^1, \ldots, M^r$, where each model assumes a different initial sensor bearing standard deviation. The filter then changes its bearing standard deviation adaptively.

To combine the state estimates of the filters, we need to compute a weight (i.e. a model probability) μ_k^j at time k associated with each model M^j. (These weights must always sum to 1 across all models.) Let the initial weights μ_0^j follow a uniform distribution: e.g. for four models, $\mu_0^1 = \mu_0^2 = \mu_0^3 = \mu_0^4 = 0.25$. The corresponding weights μ_k^j at time k can be calculated recursively using Bayes' rule:

$$\mu_k^j = \frac{p(z_k|M^j)\,\mu_{k-1}^j}{\sum_{i=1}^r p(z_k|M^i)\,\mu_{k-1}^i}\,, \tag{6.1}$$

where $p(z_k|M^j)$ is the likelihood of measurement z_k given model M^j:

$$p(z_k|M^j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp \frac{-\left(z_k - \hat{z}_{k|k-1}^j\right)^2}{2\sigma_j^2}\,, \tag{6.2}$$

where $\sigma_j^2 = \boldsymbol{H}_k^j \boldsymbol{P}_{k|k-1}^j \boldsymbol{H}_k^{j\,T} + \left(\sigma_\theta^2\right)^j$ is the innovation variance for model M^j. Here, $\boldsymbol{H}_k^j$ and $\boldsymbol{P}_{k|k-1}^j$ are the linearised measurement matrix and predicted covariance for model M^j respectively, $\left(\sigma_\theta^2\right)^j$ is the variance of the measured angle for model M^j [see (2.11)], and $\hat{z}_{k|k-1}^j$ is the predicted bearing at time k for model M^j.

The adaptive state estimate is obtained as a weighted sum of the estimates produced by all of the models:

$$\hat{\boldsymbol{x}}_{k|k} = \sum_{j=1}^r \mu_k^j\, \hat{\boldsymbol{x}}_{k|k}^j\,. \tag{6.3}$$

Similarly, the adaptive covariance matrix may be computed as

$$\boldsymbol{P}_{k|k} = \sum_{j=1}^r \mu_k^j \left[\boldsymbol{P}_{k|k}^j + \left(\hat{\boldsymbol{x}}_{k|k}^j - \hat{\boldsymbol{x}}_{k|k}\right)\left(\hat{\boldsymbol{x}}_{k|k}^j - \hat{\boldsymbol{x}}_{k|k}\right)^T\right]. \tag{6.4}$$

Figure 15 illustrates the SMM algorithm for $r = 2$ models. Implementation of the algorithm is shown in Figure 16.

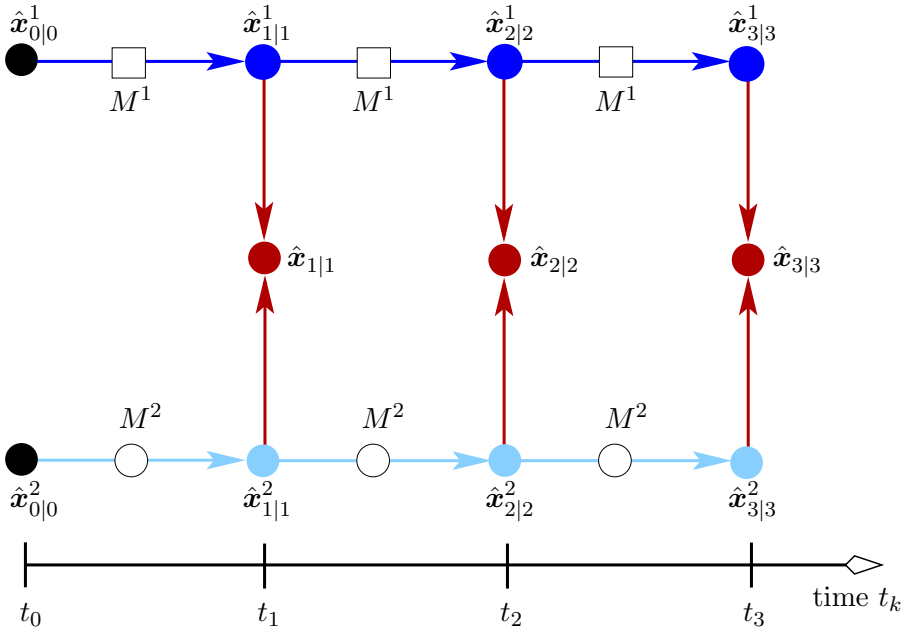


Figure 15: *Two-model example of SMM extracted from [14]. The models are denoted M^1 (dark blue path) and M^2 (light blue path). Red paths point to the final estimates that are output at each time step*

At time $k = 1, 2, \ldots,$ for each model M^j $(j = 1, \ldots, r)$:

Initialise filters at the previous time step with:

Model estimates: $\mathcal{N}\big(\hat{\boldsymbol{x}}^j_{k-1|k-1}, \boldsymbol{P}^j_{k-1|k-1}\big)$

Model probabilities: μ^j_{k-1}

Model matched filtering: For model M^j $(j = 1, \ldots, r)$,

Model conditioned Kalman filter produces $\mathcal{N}\big(\hat{\boldsymbol{x}}^j_{k|k}, \boldsymbol{P}^j_{k|k}\big)$

Likelihood for model M^j: $p(z_k|M^j) \equiv p\big(z_k|M^j, \hat{\boldsymbol{x}}^j_{k-1|k-1}, \boldsymbol{P}^j_{k-1|k-1}\big)$ calculated in (6.2)

Model probability update: $\mu^j_k = \frac{p(z_k|M^j)\,\mu^j_{k-1}}{\sum_{i=1}^r p(z_k|M^i)\,\mu^i_{k-1}}$

Combined estimate:

$\hat{\boldsymbol{x}}_{k|k} = \sum_{j=1}^r \mu^j_k\, \hat{\boldsymbol{x}}^j_{k|k}$

$\boldsymbol{P}_{k|k} = \sum_{j=1}^r \mu^j_k \left[\boldsymbol{P}^j_{k|k} + \big(\hat{\boldsymbol{x}}^j_{k|k} - \hat{\boldsymbol{x}}_{k|k}\big)\big(\hat{\boldsymbol{x}}^j_{k|k} - \hat{\boldsymbol{x}}_{k|k}\big)^T\right]$

Figure 16: *SMM algorithm*

6.2 Dynamic Multiple Models Estimator

This type of estimator models the system as evolving according to one of r possible hypotheses or models. The system may only switch between models at the discrete times $t_1, t_2, \ldots$ This switching is assumed to be Markov, so that the model in the period $(t_k, t_{k+1}]$ depends only on the model in the period $(t_{k-1}, t_k]$. For example, initially $(t = t_0)$ there is just one model. Over the time interval $(t_0, t_1]$, this single model branches r times, giving r models at time t_1. Each of these models then branches again, producing a total of r^2 hypotheses over $(t_0, t_2]$, and again a further splitting produces r^3 hypotheses over $(t_0, t_3]$.

This exponential growth in the number of models over time from a dynamic multiple models estimator is called the "hypothesis growth problem". The GPB1, GPB2, and IMM methods were proposed as a means for controlling such unwanted growth. Each of these estimators is now described in detail.

Let M_k^i denote the regime in which the system follows model M^i during the time interval $(t_{k-1}, t_k]$. We assume that the probability p_{ij} of switching from model M^i to model M^j is known and independent of the switching time: i.e. $p_{ij} \equiv p\{M_{k+1}^j | M_k^i\}$ is independent of k for all $k > 0$, and $\sum_{j=1}^{r} p_{ij} = 1$ for all $i = 1, \ldots, r$. Again, demand the weights sum to 1 and initially follow a uniform distribution (i.e. μ_0^j is constant and independent of model M^j).

6.2.1 Generalised Pseudo Bayes Methods (GPB1 and GPB2)

One of the first methods proposed for controlling the growth of hypotheses for switching systems was the generalised pseudo Bayes algorithm (GPB). In the GPB, hypotheses are merged according to the manoeuvre history of the previous time steps. If several hypotheses have the same manoeuvre history over the previous $n-1$ time intervals, then these hypotheses are merged, and the algorithm is denoted GPBn. Thus the GPBn maintains r^n hypotheses, which are collapsed to r^{n-1} hypotheses at each time step following measurement update.

- For the GPB1, r models are merged to a single component every time step after measurement update. All hypothesis histories that differ only in older models are merged. All r filters are effectively reinitialised with the merged overall estimate from the previous time step. GPB1 can be viewed as an approximation to the full Bayesian hypothesis tree multiple model filter, which considers only the possible models at the latest time instant and merges all possible model histories into one hypothesis. The cycle then repeats. Figure 17 shows an example of $r = 2$ models; Figure 18 presents the GPB1 algorithm.

- For the GPB2 an extra time step history is kept, and so r^2 models are merged to r models after each measurement update. The GPB2 considers models at the latest two time instants, and merges all previous model sequences that end at the same model. Thus the GPB2 maintains r model-matched estimates. For each of the r estimates, r filters corresponding to the possible model transitions are run (a total of r^2 filters). The updated states that end at the same model are then merged, and the cycle repeats. Figure 19 shows an example of $r = 2$ models; Figure 20 presents the GPB2 algorithm.

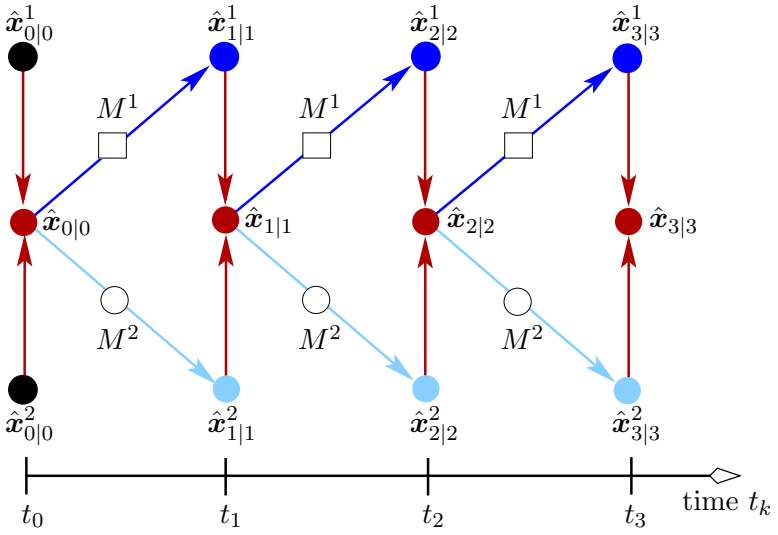


Figure 17: *Two-model example of GPB1 [19]. The colours are the same as those used in Figure 15*

At time $k = 1, 2, \ldots$:

Initialise filters at the previous time step with:

Combined estimate: $\mathcal{N}\big(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{P}_{k-1|k-1}\big)$

Model probabilities: μ^j_{k-1} for model M^j $(j = 1, \ldots, r)$

Model matched filtering: For model M^j $(j = 1, \ldots, r)$,

Model conditioned Kalman filter produces $\mathcal{N}\big(\hat{\boldsymbol{x}}^j_{k|k}, \boldsymbol{P}^j_{k|k}\big)$

Likelihood $\Lambda^j_k = p\big(z_k | M^j_k, \hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{P}_{k-1|k-1}\big)$, where M^j_k denotes the regime in which the system follows model M^j during the time interval $(t_{k-1}, t_k]$

Model probability update:

$$\mu^j_k = \frac{1}{A}\Lambda^j_k \sum_{i=1}^{r} p_{ij}\, \mu^i_{k-1}\,, \quad \text{where } A \equiv \sum_{j=1}^{r}\Big[\Lambda^j_k \sum_{i=1}^{r} p_{ij}\, \mu^i_{k-1}\Big]$$

Combined estimate:

$$\hat{\boldsymbol{x}}_{k|k} = \sum_{j=1}^{r} \mu^j_k\, \hat{\boldsymbol{x}}^j_{k|k}$$

$$\boldsymbol{P}_{k|k} = \sum_{j=1}^{r} \mu^j_k \Big[\boldsymbol{P}^j_{k|k} + \big(\hat{\boldsymbol{x}}^j_{k|k} - \hat{\boldsymbol{x}}_{k|k}\big)\big(\hat{\boldsymbol{x}}^j_{k|k} - \hat{\boldsymbol{x}}_{k|k}\big)^T\Big]$$

Figure 18: *GPB1 algorithm extracted from [14, 19]*

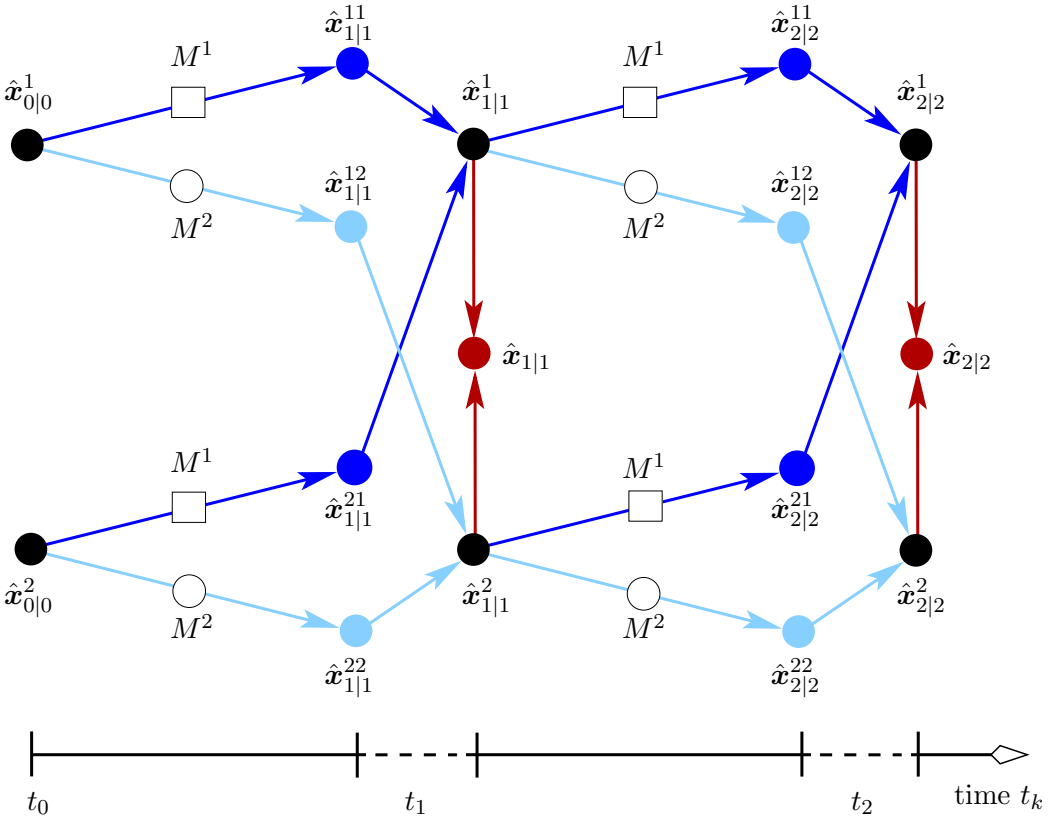


Figure 19: *Two-model example of GPB2 [19]. The colours are the same as those used in Figure 15. The dotted lines at time t_1 reflect the fact that e.g. the seven estimates with subscript "1|1" are all created in a quick succession that does not depend on the time line; they can be considered as all being created at time t_1*

At time $k = 1, 2, \ldots$, for each model M^i $(i = 1, \ldots, r)$,

Initialise filters at the previous time step with:

Model estimates: $\mathcal{N}\big(\hat{\boldsymbol{x}}^i_{k-1|k-1}, \boldsymbol{P}^i_{k-1|k-1}\big)$

Model probabilities: μ^i_{k-1}

Model matched filtering: For models M^i, M^j $(i, j = 1, \ldots, r)$,

With M^j_k denoting the regime in which model M^j applies during the time interval $(t_{k-1}, t_k]$, the model conditioned Extended Kalman filter for M^j_k takes $\mathcal{N}\big(\hat{\boldsymbol{x}}^i_{k-1|k-1}, \boldsymbol{P}^i_{k-1|k-1}\big)$ and produces $\mathcal{N}\big(\hat{\boldsymbol{x}}^{ij}_{k|k}, \boldsymbol{P}^{ij}_{k|k}\big)$.

Likelihood: $\Lambda^{ij}_k = p\big(z_k | M^j_k, \hat{\boldsymbol{x}}^i_{k-1|k-1}, \boldsymbol{P}^i_{k-1|k-1}\big)$

Model merging probabilities: With $\mu^{ij}_{k-1|k}$ the probability that M^i was in effect at time $k-1$ given that M^j is in effect at time k conditioned on the data $\boldsymbol{Z}^k$,

$$\mu^{ij}_{k-1|k} = \frac{1}{A_j} \Lambda^{ij}_k \, p_{ij} \, \mu^i_{k-1} \,,$$

where $A_j \equiv \sum_{i=1}^r \Lambda^{ij}_k \, p_{ij} \, \mu^i_{k-1}$.

Merging: For model M^j $(j = 1, \ldots, r)$,

$$\hat{\boldsymbol{x}}^j_{k|k} = \sum_{i=1}^r \mu^{ij}_{k-1|k} \, \hat{\boldsymbol{x}}^{ij}_{k|k}$$

$$\boldsymbol{P}^j_{k|k} = \sum_{i=1}^r \mu^{ij}_{k-1|k} \Big[\boldsymbol{P}^{ij}_{k|k} + \big(\hat{\boldsymbol{x}}^{ij}_{k|k} - \hat{\boldsymbol{x}}^j_{k|k}\big)\big(\hat{\boldsymbol{x}}^{ij}_{k|k} - \hat{\boldsymbol{x}}^j_{k|k}\big)^T \Big]$$

Model probability update: For model M^j $(j = 1, \ldots, r)$,

$$\mu^j_k = \frac{1}{A} \sum_{i=1}^r \Lambda^{ij}_k \, p_{ij} \, \mu^i_{k-1} = \frac{A_j}{A}$$

where $A \equiv \sum_{j=1}^r A_j$.

Combined estimate (for output):

$$\hat{\boldsymbol{x}}_{k|k} = \sum_{j=1}^r \mu^j_k \, \hat{\boldsymbol{x}}^j_{k|k} \,, \quad \boldsymbol{P}_{k|k} = \sum_{j=1}^r \mu^j_k \Big[\boldsymbol{P}^j_{k|k} + \big(\hat{\boldsymbol{x}}^j_{k|k} - \hat{\boldsymbol{x}}_{k|k}\big)\big(\hat{\boldsymbol{x}}^j_{k|k} - \hat{\boldsymbol{x}}_{k|k}\big)^T \Big]$$

Figure 20: *GPB2 algorithm extracted from [14, 19]*

6.2.2 Interacting Multiple Models Method (IMM)

The IMM is essentially an approximation to the GPB2, but only requires r filters instead of the GPB2's r^2 filters. Rather than perform the merging operation immediately after measurement update, the IMM introduces a merging procedure termed an "interaction" immediately after hypothesis branching, but before prediction. The mixture of assumed Gaussian probability densities is approximated by a single Gaussian density via moment matching. The merging process is therefore moved backwards by almost a full filter cycle. Consequently, it is no longer required to carry out the prediction and measurement update operations for r^2 branching hypotheses, but only for the r merged hypotheses. Figure 21 shows an example with $r = 2$ models; Figure 22 presents the IMM algorithm.

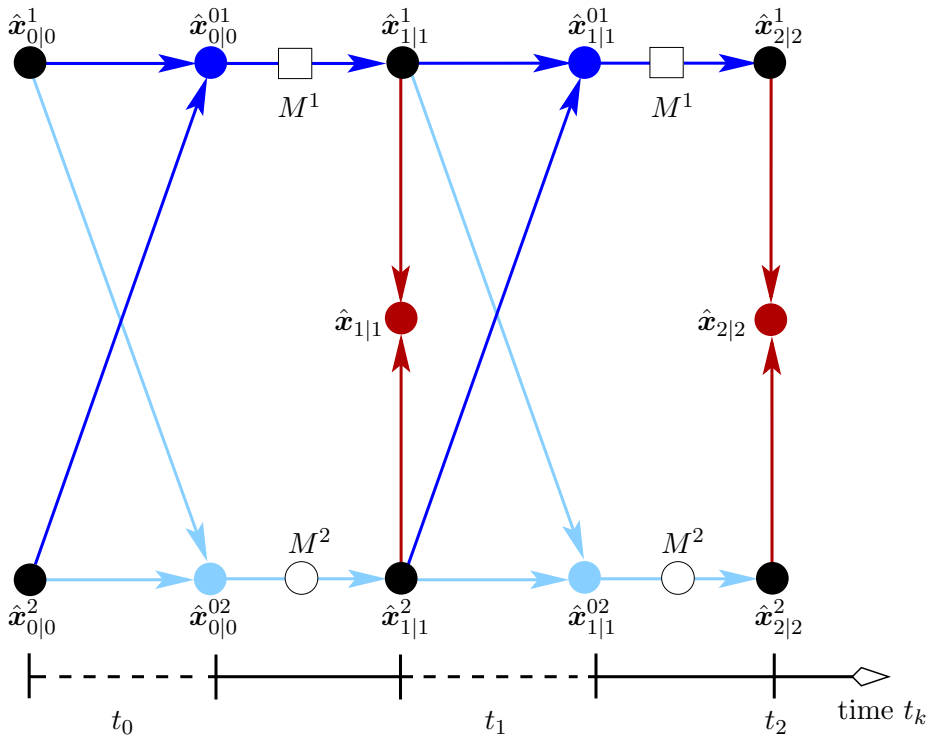


Figure 21: *Two-model example of IMM [19]. The colours and use of the dotted sections on the time axis are the same as used in previous figures*

At time $k = 1, 2, \ldots$, for model M^i $(i = 1, \ldots, r)$,

Initialise filters at the previous time step with:

Model estimate: $\mathcal{N}\big(\hat{\boldsymbol{x}}^i_{k-1|k-1}, \boldsymbol{P}^i_{k-1|k-1}\big)$

Model probabilities: $\mu^i_{k-1|k-1}$.

Mixing probabilities: For models M^i, M^j $(i, j = 1, \ldots, r)$, with $\mu^{ij}_{k-1|k-1}$ the probability that M^i was in effect at $k-1$ given that M^j is in effect at k conditioned on the data $\boldsymbol{Z}^{k-1}$ up to and including time $k-1$,

$$\mu^{ij}_{k-1|k-1} = \frac{1}{\bar{A}_j}\, p_{ij}\, \mu^i_{k-1|k-1}\,, \quad \text{where } \bar{A}_j \equiv \sum_{i=1}^{r} p_{ij}\, \mu^i_{k-1|k-1}$$

Mixing: For model M^j $(j = 1, \ldots, r)$,

$$\hat{\boldsymbol{x}}^{0j}_{k-1|k-1} = \sum_{i=1}^{r} \mu^{ij}_{k-1|k-1}\, \hat{\boldsymbol{x}}^i_{k-1|k-1}$$

$$\boldsymbol{P}^{0j}_{k-1|k-1} = \sum_{i=1}^{r} \mu^{ij}_{k-1|k-1} \Big[\boldsymbol{P}^i_{k-1|k-1} + \big(\hat{\boldsymbol{x}}^i_{k-1|k-1} - \hat{\boldsymbol{x}}^{0j}_{k-1|k-1}\big)\big(\hat{\boldsymbol{x}}^i_{k-1|k-1} - \hat{\boldsymbol{x}}^{0j}_{k-1|k-1}\big)^T\Big]$$

Model matched filtering: For model M^j $(j = 1, \ldots, r)$,

Prediction:

$$\hat{\boldsymbol{x}}^{0j}_{k|k-1} = \boldsymbol{F}^j_k\, \hat{\boldsymbol{x}}^{0j}_{k-1|k-1}$$

$$\boldsymbol{P}^{0j}_{k|k-1} = \boldsymbol{F}^j_k\, \boldsymbol{P}^{0j}_{k-1|k-1}\, {\boldsymbol{F}^j_k}^T + \boldsymbol{Q}^j_k$$

Update:

$$\hat{\boldsymbol{z}}_{k|k-1} = h_k(\hat{\boldsymbol{x}}^{0j}_{k|k-1}) \text{ as in (2.10)}$$

$$\boldsymbol{H}^j_k = \frac{\partial h_k(\boldsymbol{x})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}^{0j}_{k|k-1}}$$

$$S^j_k = \boldsymbol{H}^j_k\, \boldsymbol{P}^{0j}_{k|k-1} {\boldsymbol{H}^j_k}^T + \big(\sigma^2_\theta\big)^j_k$$

$$\boldsymbol{K}^j_k = \boldsymbol{P}^{0j}_{k|k-1} {\boldsymbol{H}^j_k}^T \big(S^j_k\big)^{-1}$$

$$\boldsymbol{v}^j_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1}$$

$$\hat{\boldsymbol{x}}^j_{k|k} = \hat{\boldsymbol{x}}^{0j}_{k|k-1} + \boldsymbol{K}^j_k\, \boldsymbol{v}^j_k$$

$$\boldsymbol{P}^j_{k|k} = \big(\boldsymbol{I} - \boldsymbol{K}^j_k \boldsymbol{H}^j_k\big)\boldsymbol{P}^{0j}_{k|k-1}$$

$$\Lambda^j_k \sim \mathcal{N}\big(\boldsymbol{v}^j_k; 0, S^j_k\big)$$

Model probability update: For model M^j $(j = 1, \ldots, r)$,
$\mu^j_{k|k} = \frac{1}{A}\Lambda^j_k\, \bar{A}_j$ where $A \equiv \sum_{i=1}^{r} \Lambda^i_k\, \bar{A}_i$.

Combined estimate (for output):

$$\hat{\boldsymbol{x}}_{k|k} = \sum_{j=1}^{r} \mu^j_{k|k}\, \hat{\boldsymbol{x}}^j_{k|k}\,, \quad \boldsymbol{P}_{k|k} = \sum_{j=1}^{r} \mu^j_{k|k} \Big[\boldsymbol{P}^j_{k|k} + \big(\hat{\boldsymbol{x}}^j_{k|k} - \hat{\boldsymbol{x}}_{k|k}\big)\big(\hat{\boldsymbol{x}}^j_{k|k} - \hat{\boldsymbol{x}}_{k|k}\big)^T\Big]$$

Figure 22: *IMM algorithm extracted from [19]*

7 Adaptive Filters Results

This section presents an analysis of whether the adaptive filters of Section 6 (SMM, GPB1, GPB2, IMM) might be of benefit when a filter lacks knowledge of the measurement noises (e.g. variance or biases). To this end, we investigate how to compute a filter bearing standard deviation adaptively.

The scenario and parameters described in Section 4 are used here (we may recall that the "true" sensor $\sigma_\theta = 1.5°$), except now the filters do not "know" this standard deviation. The parameters defined in Section 6 and used specifically for adaptive filters are summarised as follows:

- We use four models; i.e. $r = 4$ in Section 6. These models M^j, $j = 1, 2, 3, 4$, use initial sensor bearing standard deviations of 1.5°, 2.5°, 3.5°, 4.5° respectively. That is,

$$\sigma_\theta^{j=1} = 1.5°, \qquad \sigma_\theta^{j=2} = 2.5°,$$
$$\sigma_\theta^{j=3} = 3.5°, \qquad \sigma_\theta^{j=4} = 4.5°.$$

- Initial uniform weights for each model are $\mu_0^1 = \mu_0^2 = \mu_0^3 = \mu_0^4 = 0.25$, and

- $[p_{ij}] = \begin{bmatrix} 0.85 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.85 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.85 \end{bmatrix}$,

where each row (but not column) of the matrix of p_{ij} is required to sum to 1. Note that the p_{ij} are only used for the dynamic multiple models estimators: GPB1, GPB2, and IMM.

The NEES analysis at the end of Section 5.1 concluded that the CEKF and CUKF outperform the MPEKF in the presence of bearing measurement variance mismatch between the model and the system. Because of this, we have restricted the following analysis to adaptive filters using CEKFs (comparing these with the CEKF single-model result), and then adaptive filters using CUKFs (comparing these with the CUKF single-model result). We present results based on two classes of problem, as follows:

Section 7.1 deals with the static sensor performance case; hence the SMM is applicable here.

Section 7.2 deals with the dynamic sensor performance case; hence GPB1, GPB2, and IMM are applicable.

7.1 Static sensor performance case

This case deals with the SMM. We compare each standard filter (CEKF, CUKF) having a bearing standard deviation matched to that of the system (which takes the values $1.5°, 2.5°, 3.5°, 4.5°$ in turn) with the SMM (using multiple CEKFs and CUKFs). We let the SMM determine its filter bearing standard deviation adaptively (again from $1.5° \to 4.5°$). For brevity we only present the results for system bearing standard deviations of 1.5° and 2.5°, because the cases for 3.5° and 4.5° show similar trends.

A comparison of the effectiveness of the SMM with the standard filters is made as follows. For each of the system bearing standard deviations, we plot

Figure 23: model probabilities versus time (averaged over 100 Monte Carlo runs),

Figure 24: RMS position error versus time,

Figure 25: RTAMS position error (an average over the 18 → 30 minutes interval) versus filter type,

Figure 26: norm of bias position error at 30 minutes versus filter type, and

Figure 27: $\overline{\text{NEES}}$ versus time.

The graph legends are annotated as follows:

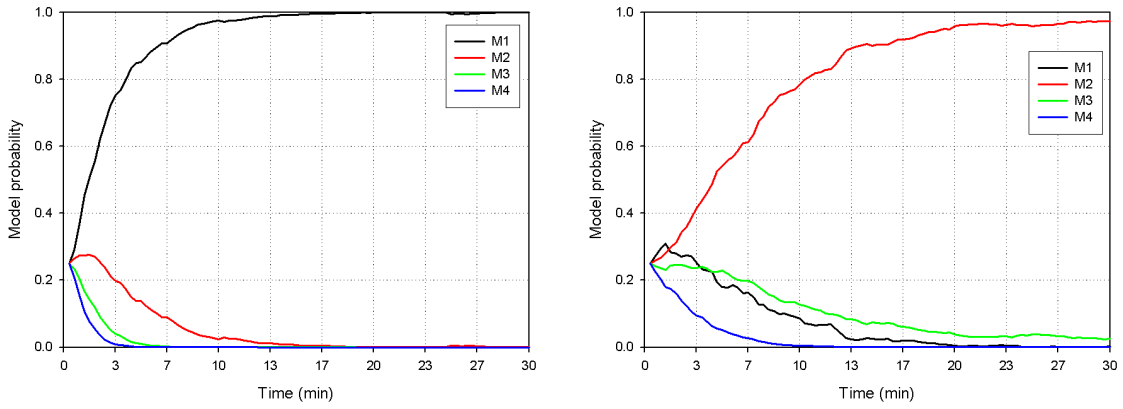
CEKF denotes the standard CEKF,

SMM-CEKF denotes the adaptive filter SMM (using multiple CEKFs),

CUKF as for "**CEKF**" above, but now using the CUKF, and

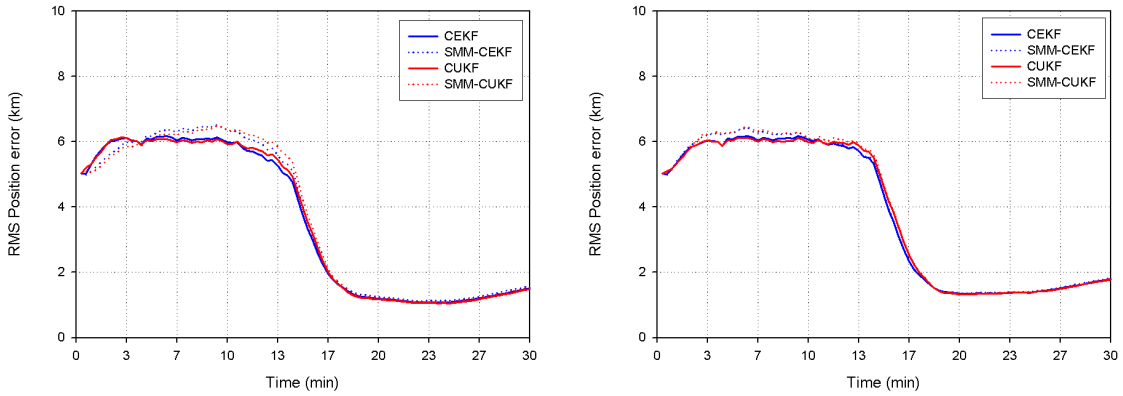
SMM-CUKF as for "**SMM-CEKF**" above, but now using multiple CUKFs.

The results show that the SMM can estimate its bearing standard deviation adaptively; that is, it matches the value to that of the system (Figure 23 shows higher probability for this case). The SMM performance results are comparable to those of the filter that uses a bearing standard deviation matched to the system. The CUKF performs slightly better than the CEKF. Also, most of the $\overline{\text{NEES}}$ values fall inside the acceptable χ^2 bounds.



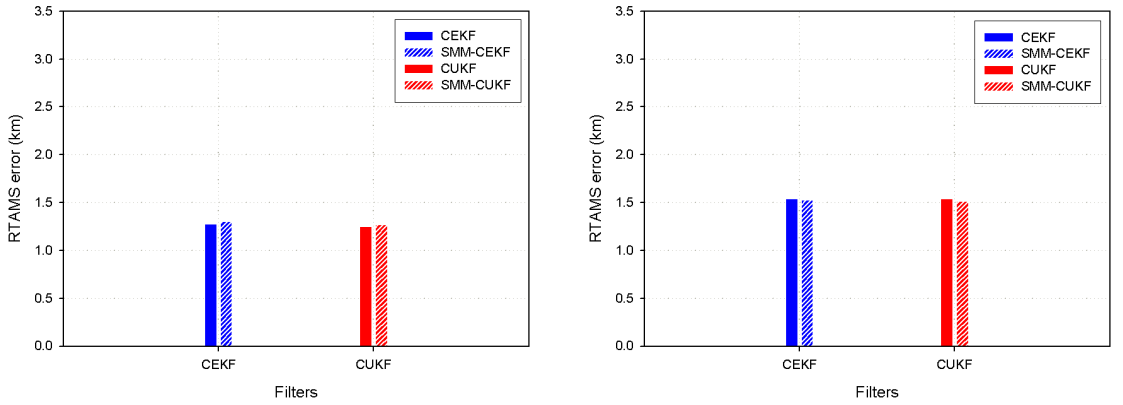
(a) System $\sigma_\theta = 1.5°$ (b) System $\sigma_\theta = 2.5°$

Figure 23: *SMM-CEKF model probabilities versus time for two system bearing standard deviations, averaged over 100 Monte Carlo runs. The SMM-CUKF shows similar trends and so is not shown here; M1–M4 denote models with bearing standard deviations of 1.5°, 2.5°, 3.5°, 4.5° respectively. In both plots, the model whose bearing standard deviation matches that of the system soon becomes dominant, showing that the filter successfully matches its bearing error to that of the system*



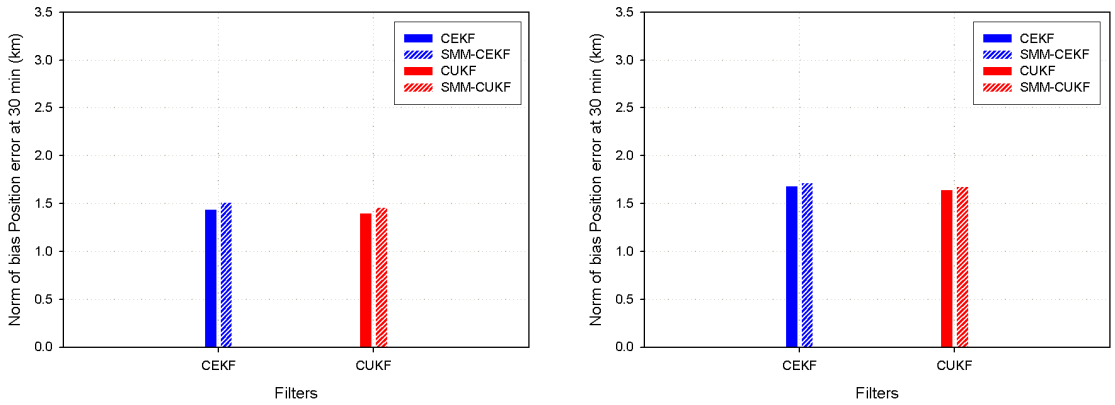
(a) System $\sigma_\theta = 1.5°$ (b) System $\sigma_\theta = 2.5°$

Figure 24: *RMS position error versus time for two system bearing standard deviations*



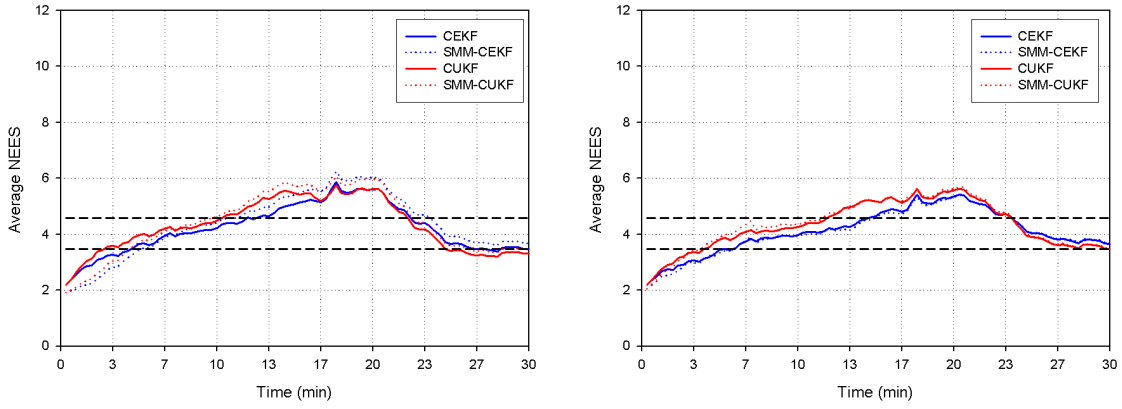
(a) System $\sigma_\theta = 1.5°$ (b) System $\sigma_\theta = 2.5°$

Figure 25: *RTAMS position error versus filter type for two system bearing standard deviations*



(a) System $\sigma_\theta = 1.5°$ (b) System $\sigma_\theta = 2.5°$

Figure 26: *Norm of bias position error at $t = 30$ minutes versus filter type for two system bearing standard deviations*



(a) System $\sigma_\theta = 1.5°$ (b) System $\sigma_\theta = 2.5°$

Figure 27: *$\overline{NEES}$ versus time for two system bearing standard deviations. The horizontal black dashed lines across are the acceptable χ^2 bounds*

7.2 Dynamic sensor performance case

Here we consider the dynamic multiple models estimators: GPB1, GPB2, IMM. The dynamic sensor performance is based on using two fixed bearing standard deviations, one for each half of the scenario. We consider two cases:

- The first half scenario uses 1.5° and the second half scenario uses 2.5° standard deviation; and
- The first half scenario uses 2.5° and the second half scenario uses 3.5° standard deviation.

For each case of the dynamic sensor performance, we compare each ideal filter (using CEKF—having a bearing standard deviation matched to that of the system—with the adaptive filters (using multiple CEKFs).[2] The adaptive filters determine their own best value of the filter bearing standard deviation from the set $\{1.5°, 2.5°, 3.5°, 4.5°\}$.

In order to compare the effectiveness of the adaptive filters with each ideal filter, we plot

Figure 28: model probabilities versus time (averaged over 100 Monte Carlo runs),

Figure 29: RMS position error versus time,

Figure 30: RTAMS position error (an average over the $18 \rightarrow 30$ minutes interval) versus filter type,

Figure 31: norm of bias position error at 30 minutes versus filter type, and

Figure 32: $\overline{\text{NEES}}$ versus time.

The graph legends use the following terms:

Ideal denotes the ideal filter (using CEKF),

GPB1 denotes the adaptive filter GPB1 (using multiple CEKFs),

GPB2 as for GPB1 above, but now using the adaptive filter GPB2 (with multiple CEKFs), and

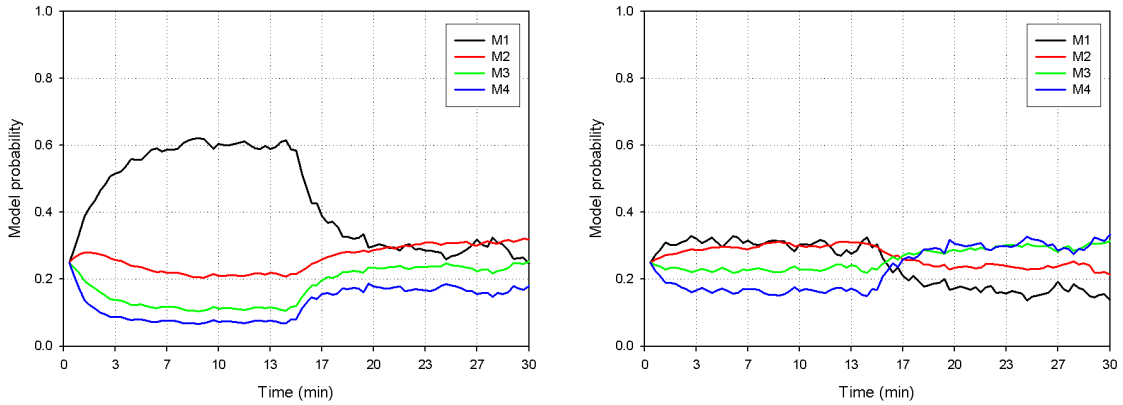
IMM as for GPB1 and GPB2 above, but now using the adaptive filter IMM (with multiple CEKFs).

Our results indicate that the adaptive filters (GPB1, GPB2, IMM) can determine their own bearing standard deviation with reasonable success. For example, Figure 28 shows that the model weights for bearing errors equal to those of the system do attain some dominance over competing weights over the course of the scenario, although this dominance is not strong. That is, in Figure 28(a) we expect model M^1 to become dominant in the first half of the scenario, with model M^2 dominating in the second half. This is mostly what happens, although the model M^2 domination is not overly pronounced. In Figure 28(b) we expect model M^2 to become dominant in the first half of the scenario, followed by

[2] Analogous results for the CUKF show similar trends to those of the CEKF and so are not presented here.

model M^3 in the second half. The effect is less obvious here, and model M^4 also becomes prominent.

These filters show performance comparable to that of an ideal filter that uses the same dynamic bearing standard deviations as those of the system. Figure 29 shows that the IMM and GPB2 perform marginally more closely to the ideal filter than does the GPB1. The GPB1 may be marginally better than the GPB2 and the IMM, giving slightly smaller errors in Figures 29–31. Figure 32 shows that the state covariances calculated using the IMM and GPB2 are more acceptable under the χ^2 test for consistency (i.e. more $\overline{\text{NEES}}$ fall inside the acceptable χ^2 bounds), and are closer to those of an ideal filter. The computer processing times required for running the IMM and GPB1 algorithms are roughly similar, and are about half of the time required to run the GPB2 routine. For these reasons, the IMM seems slightly more promising to use.



(a) System $\sigma_\theta = 1.5°$ then $2.5°$, GPB1

(b) System $\sigma_\theta = 2.5°$ then $3.5°$, GPB1

Figure 28: *Model probabilities versus time for two dynamic system bearing standard deviations: (a) 1.5° for the first half scenario and 2.5° for the second half scenario (left), (b) 2.5° for the first half scenario and 3.5° for the second half scenario (right) (GPB1 using CEKF only; the GPB2 and IMM show similar trends and so are not shown here); M1–M4 denote models with bearing standard deviations of 1.5°, 2.5°, 3.5°, 4.5° respectively.*

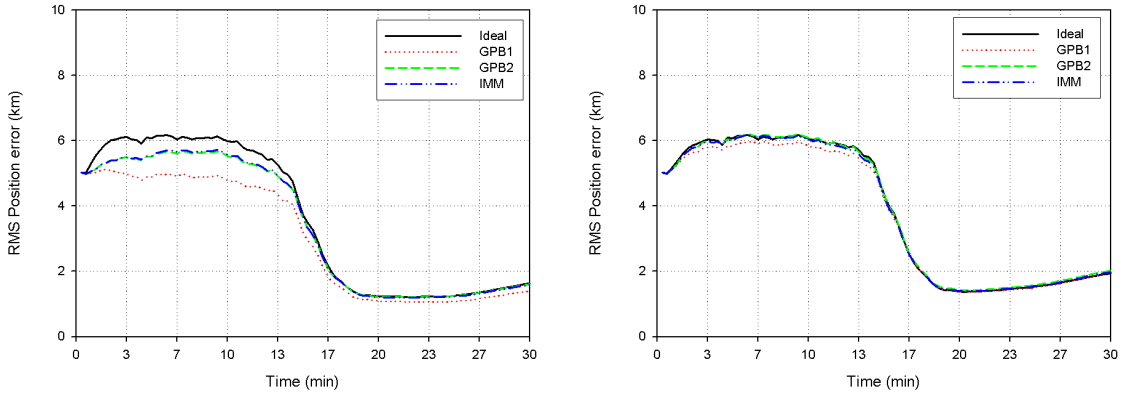


Figure 29: *RMS position error versus time for each system with dynamic bearing standard deviations: (a) 1.5° for the first half scenario and 2.5° for the second half scenario, (b) 2.5° for the first half scenario and 3.5° for the second half scenario (CEKF only)*

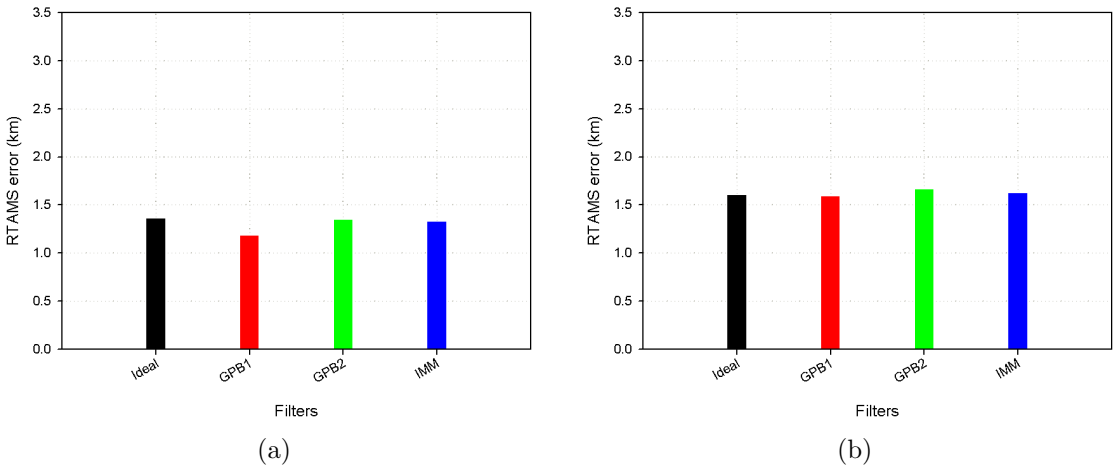


Figure 30: *RTAMS position error (averaged over $t = 18 \rightarrow 30$ minutes) for all filters, for each system with dynamic bearing standard deviations: (a) 1.5° for the first half scenario and 2.5° for the second half scenario, (b) 2.5° for the first half scenario and 3.5° for the second half scenario (CEKF only)*

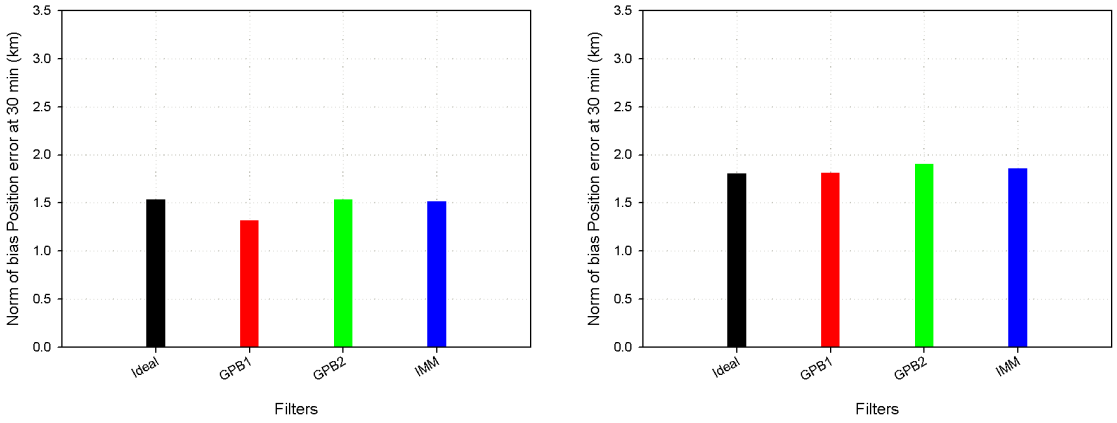


Figure 31: *Norm of bias position error at 30 minutes for all filters, for each system with dynamic bearing standard deviations: (a) 1.5° for the first half scenario and 2.5° for the second half scenario, (b) 2.5° for the first half scenario and 3.5° for the second half scenario (CEKF only)*

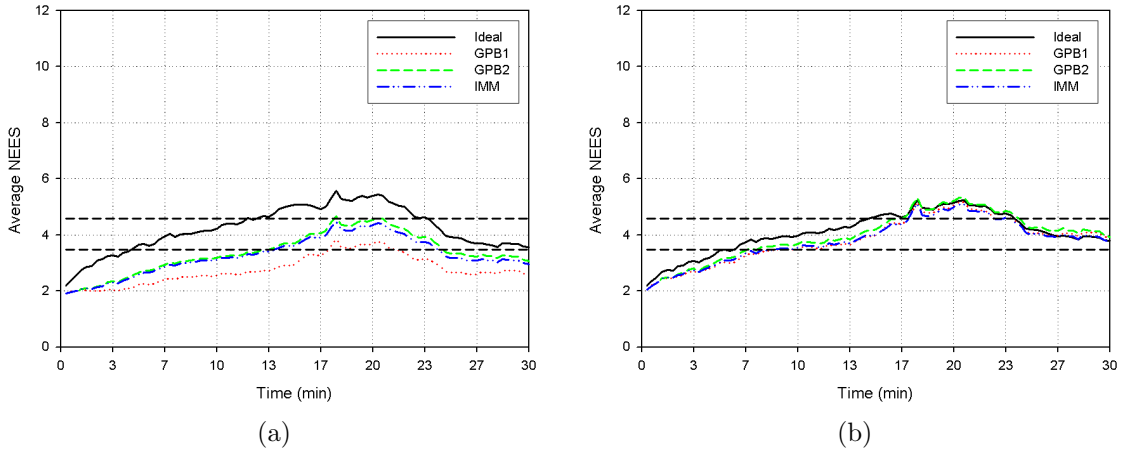


Figure 32: *$\overline{NEES}$ versus time for each system with dynamic bearing standard deviations: (a) 1.5° for the first half scenario and 2.5° for the second half scenario, (b) 2.5° for the first half scenario and 3.5° for the second half scenario (CEKF only). The horizontal dashed lines are the acceptable χ^2 bounds*

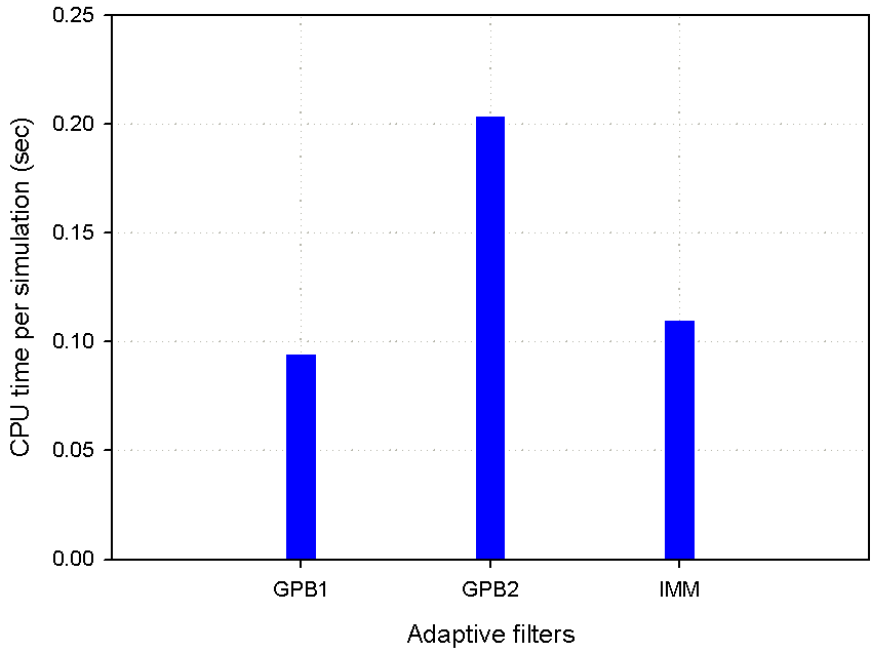


Figure 33: *CPU time per simulation run for the adaptive filters (CEKF only)*

8 Conclusion and Future Directions

In this report we have presented a benchmark of existing bearings-only TMA algorithms that are designed to cope with model uncertainty. The standard algorithms used were CEKF, MPEKF, CUKF. The main thrust of the report was the design of algorithms to cope with models mismatch: the *adaptive filters* SMM, GPB1, GPB2, IMM. These filters were designed to be able to estimate and adjust the measurement noise parameters as new data comes in.

In order to estimate the target state accurately, the estimator needs to know the statistical characteristics of the measurement noise: its bias (mean) and spread (variance). But this information tends to be blurred by factors such as frequency, signal-to-noise ratio (SNR), bearing, bearing rate, and array type, so filters using a wrong bearing mean or variance can give very poor TMA. We have concentrated on finding the variances of measurement noises adaptively. The adaptive filters show promise in being able to determine what bearing variance they should use. We stress that this is a theoretical investigation, and propose to investigate further using real data.

We are also concerned solely with single sensor/single non-manoeuvring target tracking (e.g. tracking a ship or a submarine). Both static and dynamic sensor performances were considered. For simplicity, the measurement noise used was restricted to Gaussian.

Our key results are summarised as follows. They have been tested on three scenarios encompassing low to high rates of bearing change:

- The CUKF and CEKF perform better than the MPEKF for our scenarios;
- Filters incorporating approximately the same bearing mean and error as those of the system outperform filters that are not matched in such a way;
- Adaptive filters (that tune their own model of the bearing variance) show promise for use.

Further work can be done in the following areas:

- As stated above, this work is a theoretical investigation. Future work aims to validate using real data.
- We can model the dynamic sensor performance to be more realistic; e.g. rather than use two fixed bearing standard deviations for each half of the scenario, we can make the bearing standard deviation change progressively with time. This could be implemented with a random walk model.
- We can also use the SNR to model the fluctuating bearing error distribution during the simulation; e.g. a lower SNR suggests a long-range target and a higher SNR suggests a short-range target. This information allows compilation of a look-up table of standard deviation versus time.
- Non-Gaussian measurement noise can be modelled.
- Finally, the work with adaptive filters can be extended to the case of a manoeuvring target.

9 Acknowledgements

We wish to thank Mr Warren Smith, Dr Derek Bertilone, Dr Dragana Carevic, Ms Fiona Gianoli, Mr Michael Beard and Ms Cheryl Smith-Gander for assistance in the sonar areas and scenario parameter settings. Especial thanks go to Dr Sanjeev Arulampalam for his valuable comments, discussions, encouragement and critical reading of the manuscript. Finally, but not least, we also wish to thank Dr Mark Rutten (Intelligence, Surveillance and Reconnaissance Division—ISRD) for kindly spending his time reviewing this report.

References

1. AFTP 20 (2004), *Australian Submarine Tactical Instructions (U)*, SECRET AGAO.

2. S. Julier, J.K. Uhlmann (1997), "A New Extension of the Kalman Filter to Nonlinear Systems", *SPIE AeroSense Symposium/Defense Sensing, Simulations and Controls.*

3. G. Welch, G. Bishop (2006), *An Introduction to the Kalman Filter*, TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill.

4. P.S. Maybeck (1994), *Stochastic Models Estimation and Control, Vol. 1*, Navtech Book and Software Store, New York.

5. B.D.O. Anderson, J.B. Moore (1979), *Optimal Filtering*, Prentice-Hall.

6. S. Stergios (2001), *Advanced Signal Processing Handbook: Theory and Implementation for Radar, Sonar, and Medical Imaging Real-Time Systems*, CRC Press, Boca Raton, London.

7. A. Zaknich (2005), *Advanced Textbooks in Control and Signal Processing: Principles of Adaptive Filters and Self-learning Systems*, Springer Publishing.

8. S. Arulampalam (2000), *A Comparison of Recursive Style Angle-only Target Motion Analysis Algorithms*, DSTO Technical Report DSTO–TR–0917.

9. J.C. Hassab (1989), *Underwater Signal and Data Processing*, PhD thesis, CRC Press, Boca Raton, Florida, USA.

10. V.J. Aidala, S.E. Hammel (1983), "Utilization of modified polar coordinates for bearings-only tracking", *IEEE Trans. Automatic Control*, **28** (3), 283–294.

11. B. Ristic, S. Arulampalam, N. Gordon (2004), *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House.

12. S. Julier, J. Uhlmann, H.F. Durrant-Whyte (2000), "Technical Notes and Correspondence: A new method for the nonlinear transformation of means and covariances in filters and estimators", *IEEE Trans. Automatic Control*, **45**, 3.

13. R.W. Osborne III, Y. Bar-Shalom (2006), "Radar Measurement Noise Variance Estimation with Targets of Opportunity", *IEEE Aerospace Conference.*

14. Y. Bar-Shalom, X.R. Li, T. Kirubarajan (2001), *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons.

15. A. Chapman, T.W. Josefsson (1999), *ASWEX 98, Initial Analysis and Reconstruction (U)*, DSTO Technical Report DSTO-TR-0967, CONFIDENTIAL.

16. RAN web page `http://www.navy.gov.au`

17. S. Arulampalam, M. Clark, R. Vinter (2007), "Performance of the Shifted Rayleigh Filter in Single-sensor Bearings-only Tracking", *Proc. of Fusion 2007*, Canada.

18. V.J. Aidala (1979), "Kalman Filter Behavior in Bearings-Only Tracking Applications", *IEEE Trans. Aerospace and Electronic Systems*, **15**, 1.

19. B. Ristic, N. Gordon (2005), *Multisensor Tracking and Identification*, CSSIP.

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. CAVEAT/PRIVACY MARKING

2. TITLE	3. SECURITY CLASSIFICATION
Investigation of Target Motion Analysis in the Presence of Model Uncertainty	Document (U) Title (U) Abstract (U)
4. AUTHORS	5. CORPORATE AUTHOR
T.Q.S. Truong and D. Koks	Defence Science and Technology Organisation PO Box 1500 Edinburgh, SA 5111, Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO–TR–2405	AR–014–751	Technical Report	May, 2010

8. FILE NUMBER	9. TASK NUMBER	10. SPONSOR	11. No. OF PAGES	12. No. OF REFS
2009/1106908/1	07/093	DGSM	38	19

13. URL OF ELECTRONIC VERSION	14. RELEASE AUTHORITY
http://www.dsto.defence.gov.au/corporate/reports/DSTO–TR–2405.pdf	Chief, Maritime Operations Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. DSTO RESEARCH LIBRARY THESAURUS

Kalman filters
adaptive filters
target tracking
tracking filters
filters

19. ABSTRACT

This paper presents the results of an investigation of target motion analysis algorithms that are designed to cope with model uncertainty. First, some standard recursive algorithms such as the cartesian extended Kalman filter, modified polar extended Kalman filter, and cartesian unscented Kalman filter are applied to a target motion analysis problem with model uncertainty, in order to analyse the robustness of such algorithms in these conditions. Next, some adaptive algorithms are investigated. They are the static multiple models and the dynamic multiple models estimators, namely: two generalised pseudo Bayes methods and the interacting multiple model method. In this paper, the problem is restricted to a single sensor and a single non-manoeuvring target that travels at constant velocity. Both static and dynamic sensor performances are considered. For simplicity, only Gaussian measurement noise is considered. Adaptive filters are shown to have promise: they can establish a useful bearing standard deviation adaptively and robustly.